Real-Time Context-Aware Computing with Applications in Civil Infrastructure Systems

by

Manu Akula

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering)
in The University of Michigan
2013

To My Grandfather,
For bestowing upon me, the gift of curiosity.

To My Parents,
That if it were not for their sacrifice and support,
I could not achieve my success.

To My Cousins,
For their support in difficult times,
And for inspiring me to never give up.

Acknowledgments

This dissertation is a culmination of a perfect working relationship with my advisor, Dr. Vineet Kamat, to whom I will always be indebted. I have found a friend in him who views my effort sympathetically. His friendship, wisdom, and unreserved support during my PhD has generously paved the way for my accomplishments. He has been a constant source of inspiration and the best advisor and mentor a student could have.

I would also like to acknowledge the contributions of my committee members Dr. Atul Prakash, Dr. Jerry Lynch, and Dr. SangHyun Lee, each of whom significantly contributed to my graduate education in many ways. Their invaluable insight, guidance, and feedback have helped me shape the implementation of various components of my research.

I would like to thank Dr. John Everett for his insightful lectures covering the fundamentals of construction engineering and project management. I am grateful to Dr. Kristen LeFevre and Mr. Andrew Morgan, whose sincerity and patience helped me to learn the basics of database systems and programming, respectively. I would also like to thank Dr. Photios Ioannou, for always keeping an open door and dispensing invaluable advice.

I am also greatly indebted to the many people who, in some way, contributed to the progress and publication of the work contained herein. First and foremost, I would like to particularly thank Mr. Atul Sandur for his invaluable collaboration and help at a crucial juncture in my PhD journey. I also appreciate the help provided by Ms. Yilan Zhang, Dr. Gwen van der Linden, Dr.

Sharada Alampalli, Mr. Sean O'Connor, and Dr. Mohammed Ettouney for their insightful comments and suggestions that have contributed towards the completion of this dissertation. I want to express my gratitude to Dr. Kamel Saidi and Ms. Gerry Cheok for supporting me during my time at the National Institute of Standards and Technology, Gaithersburg, MD. I would also like to thank Dr. Robert Lipman, Dr. Marek Franaszek, and Dr. Mani Golparvar-Fard for helping me accomplish the research conducted at NIST.

I want to thank the wonderful companionship of my friends and colleagues Sanat Talmaki, Ali Golabchi, Shamin Asaikar, Birjodh Tiwana, Suyang Dong, Chen Feng, Nicholas Fredricks, Kyle Anderson, Seungjun Ahn, Ehsan Rezazedeh Azar, and many others. I can only think back upon the last few years with feelings of fondness and reminisce. I wish you all the very best of luck in your endeavors.

Most importantly, I would like to thank my family for providing me with unconditional love, affection, and emotional strength that served as a secure anchor during the tough and easy times. I strongly believe if it were not for their encouragement, I could not achieve any progress in my life, education, and career.

Table of Contents

List of Tables

List of Figures

List of Appendices

Abstract

This dissertation contributes a structured understanding of the fundamental processes involved in developing context-aware computing applications for the civil infrastructure industry. The civil infrastructure industry is characterized by mobile human user and machine agents actively engaged in real-time decision-making tasks in a dynamic, unstructured and evolving workspace environment. This distinguishes context-aware computing from other computing technologies in three aspects: 1) it has the ability to perceive, interpret, and adapt to the agent's evolving workspace; 2) It streamlines project data and presents the agent with information pertinent to its context, thus eliminating the agent's tasks to accomplish the same; 3) By leveraging contextual information, it supplements decision-making tasks in real-time.

This research has successfully investigated technical approaches to address fundamental aspects of introducing context-aware applications to civil engineering, including: the ubiquitous localization of mobile human and machine agents in dynamic, unstructured environments; abstraction of the spatial-context and identifying the objects of interest to the agent; and the suitability of using standard product models and databases to manage and organize data for context-aware computing applications. A computational framework for designing context-aware applications to support real-time decision-making has also been implemented. The framework allows researchers and other end users to leverage currently available context-sensing technology to design and implement innovative solutions to domain specific problems.

The researched methods have been validated through several experiments conducted at the University of Michigan, the National Institute of Standards and Technology (NIST), and the Michigan Department of Transportation (MDOT). These experiments have resulted in the implementation of several applications – to support real-life decision-making tasks – that not only serve to illustrate the usefulness of the framework, but also have significant social and economic implications. Among these applications are the controlled drilling system that warns drilling personnel when the drill bit tip is about to strike rebar or utility lines, thus helping preserve the structural integrity of concrete decks and preventing utility strike accidents; an automated fault detection system that diagnoses faulty components of an underperforming Heating, Ventilation, and Air Conditioning (HVAC) distribution network; and an innovative bridge inspection system that supports condition assessment decision-making, thus introducing objectivity to visual condition assessment by providing concurrence with the Structural Health Monitoring (SHM) data.

Chapter 1

Introduction

1.1 Importance of the Research

The civil infrastructure industry is characterized by mobile agents sensing workspace environments, analyzing large volumes of acquired and apriori information, making decisions in real-time, and communicating streamlined information with each other, all in order to accomplish their assigned tasks. This process is complicated by the unique nature of civil infrastructure projects that require tasks to be performed in a dynamic, unstructured, and evolving workspace environment (Son et al 2008). Context-aware computing has the potential to help improve the accuracy and efficiency of performing these tasks by providing decision-making support to mobile agents through automation.

Field personnel spend a significant amount of time in manually accessing the information required to support important decision-making tasks leading to a loss in productivity and money in the civil infrastructure industry (Aziz et al 2005). For example, it has been estimated that as much as 30% of project costs are wasted in the construction industry (Zou et al 2006) due to 1) unreliable information documentation, access and delivery 2) ineffective communication and 3) inefficient operation organization (Fathi et al 2006). Context-aware computing is a promising technology that can reduce inefficiencies by addressing the aforementioned issues by allowing for the documentation, organization, access and retrieval of streamlined information relevant to the task and context at hand.

Context-aware computing has been defined as the use of environmental characteristics – such as an agent’s location, identity, profile, time, and activity – in order to communicate information that is relevant to the agent’s context (Burrell and Gay 2001). Context-aware decision-making applications are implemented using a mobile distributed computing system – a collection of mobile and stationary computing devices that are cooperating and communicating on the agents’ behalf (Schilit et al 1994). These applications examine and react to the agent’s context based on information that can be used to characterize the current situation of the agent and can thus, potentially, enable mobile agents to leverage knowledge about various contextual parameters to ensure they get highly specific and streamlined information relevant to the decisions at hand (Schilit et al 1994).

It is therefore important to examine the anatomy of context-aware applications and to understand the key drivers involved in designing and developing such decision-making applications for the civil infrastructure domain. It is also important to understand the ability and limitations of currently available technologies in sensing and interpreting relevant contextual parameters. Finally, it is important to understand the implications and challenges in developing context-aware decision-making applications in the civil infrastructure domain.

1.2 Background of the Research

In information-intensive industries – such as the medical, pervasive gaming, guided tourism, automotive, and the civil infrastructure industries – rapid and convenient access to information is crucial in making critical real-time decisions. In recent years, several research prototypes and commercial applications, shown in Figure 1.1, have been developed that utilize contextual information to enhance the human experience in the aforementioned industries, and in other closely related industries (Chen and Kotz 2000).

I am leading you away from the Shire Hall entrance to Lancaster Castle. To see this route as a map click here.

Shire Hall was built in 1780 and this is now the main door to Lancaster Castle and is the starting point of the Castle tours.

Your location cell: Lancaster Castle - Shire Hall Entrance

Status: Currently receiving location updates

Traffic information is available for your route.
5.2 km +19 min
Diversion in 5.2 km
2:20 11 km
-15 min +0.8 km

Figure 1.1: Examples of context-aware applications in various domains (Ordonez et al 2009; Thomas et al 2000; Davies et al 1999; Bachmann et al 2000)

1.2.1 Previous Work in Context-Aware Decision-Making

1.2.1.1 Healthcare and Medical Industry

Modern hospitals, with clinicians moving around between different work settings and being engaged in many parallel activities, are seen as ideal scenarios for improvisation through context-aware computing applications. An ontological context management system, that defines context using terms from the medical field, has been developed to improve the adoption of mobile health information systems (Jahnke et al 2005). Traditionally, healthcare sector fundamentals (doctors and nurses) have been hesitant to embrace electronic clinical information systems in general and mobile technologies in particular. However, context-aware computing–based applications such as robotic surgery (Ordonez et al 2007), context-aware electronic patient records, context-aware hospital beds and a context-aware pill container have been evaluated and it has been concluded that making clinical applications context-aware is a key ingredient in creating more useful computer support for clinical work enabling computers in hospitals to move out of the office and into clinical work (Bardram 2004).

1.2.1.2 Guided Tourism Industry

Context-aware computing has been used in developing tourist guidance devices that provide tourists with ubiquitous and personalized access to tourism information. Currently, there is a proliferation of mobile tourist guides in use by different organizations with diverse functionalities (Schwinger at al. 2009). Such applications allow users in a nomadic environment to access services similar to those accessible through desktop computers but can be accessed anywhere, at anytime using a variety of devices (Laukkanen et al 2002). The Cyberguide project developed a prototype of a mobile tour guide that provides information to tourists based on knowledge of their position, orientation, and time. The location and orientation information is collected by Global Positioning System (GPS) in the outdoor version and by an infrared positioning system in the indoor version (Abowd et al 1997). The GUIDE project developed a context-sensitive tourist guide for visitors to the city of Lancaster where visitors were equipped with portable GUIDE units that provide interactive services and streamlined information using the city's wireless infrastructure (Cheverst et al 2000; Davies et al 1999).

Context-aware computing has also been implemented in developing applications for guidance in libraries—for example the SmartLibrary software that helps users search for and locate books in the University of Oulu by providing map-based guidance on a personal digital assistant (PDA) (Aittola et al 2003). Applications have also been developed for museums such as the Exploratorium, an interactive science museum in San Francisco (Fleck et al 2002). Several context-aware shopping assistants that guide shoppers through stores, provide details of items, help locate items, point out items on sale, and do a comparative price analysis have also been developed (Asthana et al 1994; Black et al 2009; Zhu et al 2004).

1.2.1.3 Automotive Industry

Commercially available GPS-based navigation devices for automobiles utilize contextual information—such as the automobile's location, speed, desired destination, road maps, and live traffic reports—to provide intelligent guidance for automobile drivers. BMW's ConnectDrive project combines contextual information of an automotive framework acquired from the vehicle environment and the driving state parameters (Bachmann et al 2000). The BMW SURF project integrates the ConnectDrive framework with another contextual parameter—the driver information—to create exemplary vehicle intelligence (Hoch et al 2007).

1.2.1.4 Emergency Responders and Military

Emergency responders, security, and military personnel also stand to gain significantly from context-aware applications that enhance the performance of mission critical tasks. Prototype context-aware communication applications to support search and rescue fire fighting scenarios have been developed using the Siren system (Jiang et al 2004). Investigation tasks, such as fraud detection, also stand to benefit from intelligent information systems that build and retrieve investigative context to fulfill tasks more effectively (Wen et al 2007).

Military personnel are increasingly challenged with information overload and the correlation of information from several heterogeneous sources. Infrastructure for context-aware informational retrieval that enhances situational awareness has been developed by using workflow, context, ontology, profile, and information aggregation. The developed infrastructure provides military personnel with customized, mission-oriented systems that give access to streamlined information from heterogeneous sources in the context of the mission (Bahrami et al 2007).

1.2.1.5 Office and Workspace Management

In workspaces where human and computing agents are highly mobile, context-awareness has been used to ensure that telephone (Want et al 1992) and pager (Brown et al 1997) communication reaches the appropriate personnel by forwarding messages to destinations nearest to the mobile human. An office assistant that manages employee schedules based on contextual information of the new task—such as task identity, nature, and employee schedule status—has been implemented (Yan and Selker 2000). The ComMotion (Marmasse and Schmandt 2000), Rome (Huang et al 1999), CybreMinder (Dey and Abowd 2000), and MemoClip (Beigl 2000) projects have created variations of location-aware reminder delivery systems that utilize agents' location and time parameters. These systems generate reminder messages with a location tag, and when the intended recipient arrives at that location, the message is delivered by voice synthesis.

1.2.1.6 Mobile and Smartphone Technology Industry

In the MUSA-Shadow project, internet-enabled mobile phones were integrated with a context-aware system that allowed for the development of an intelligent application in distress scenarios (Fischmeister et al 2002). The development and popularity of smartphone technology has provided the civilian population with a malleable PDA that can be customized to accommodate a variety of context-aware computing applications.

The Sleep Cycle Alarm clock is one such context-aware application that uses the accelerometer in the smartphone to monitor movement and determine the user's sleep phase (SleepCycle 3.0 2013). The bio-alarm clock analyzes sleep patterns and wakes the user up when the user is in the lightest sleep phase, creating the feeling of waking without an alarm clock—a natural way to wake up where the user feels rested and relaxed.

Another application—the Yelp Monocle—is a context-aware application that helps users access information regarding Yelp-reviewed establishments. The application uses the positioning system and the accelerometers in a smartphone to determine Yelp-reviewed establishments such as restaurants and subway stations located in the direction that the smartphone is pointed in. The application overlays establishment information onto the live camera feed, letting the user know exactly where and how far away the establishments are. The user can access the information embedded within the icons, such as Yelp reviews and menus, by clicking on the respective icon of the establishment of choice (Gizmodo 2013). Several other such proprietary context-aware applications have also recently been developed by the smartphone community and are commercially available in the market.

1.2.2 Context-Aware Decision-Making in Civil Infrastructure Industry

Domains such as maintenance, repair, construction, and manufacturing are in need of effective communication and collaboration and are considered typical 'wearable' domains. Wearable computing systems that provide users with automatic, context-sensitive information access along with interpersonal communication and collaboration have been investigated over the years (Korutem et al 1999; Pascoe et al 1998). Context-aware computing research in the civil infrastructure industry has largely been focused upon developing architecture for mobile distributed computing systems that involve mobile human users, sensing, and computing agents. Previous work in context-aware computing in the civil infrastructure industry has involved evaluating the use of wireless technologies on construction sites for delivering context-specific information on construction sites (Behzedan et al 2008; Khoury and Kamat 2009a).

Prototype applications for context-aware information delivery based on wireless network communication have been developed and are shown in Figure 1.2. One such application uses

Wireless Local Area Networking (WLAN) communication and employs a layer of context-aware intelligence to provide on-site workers with relevant tools and streamlined information to perform critical tasks more effectively (Aziz et al 2006). ZigBee—a low-cost, low-power, wireless mesh network standard—networks have been used to develop prototype applications for tracking objects on construction jobsites to provide insight into industrial practices (Skibniewski and Jang 2006). Ultra-Wide Band (UWB) wireless sensing technology has also been used to develop the capacity to determine three-dimensional resource location information on construction job sites characterized by object-cluttered environments (Teizer et al 2008). Intelligent wireless web services have also been used to develop a context-aware material delivery system in the construction logistics supply chain domain (Omar and Ballal 2009).

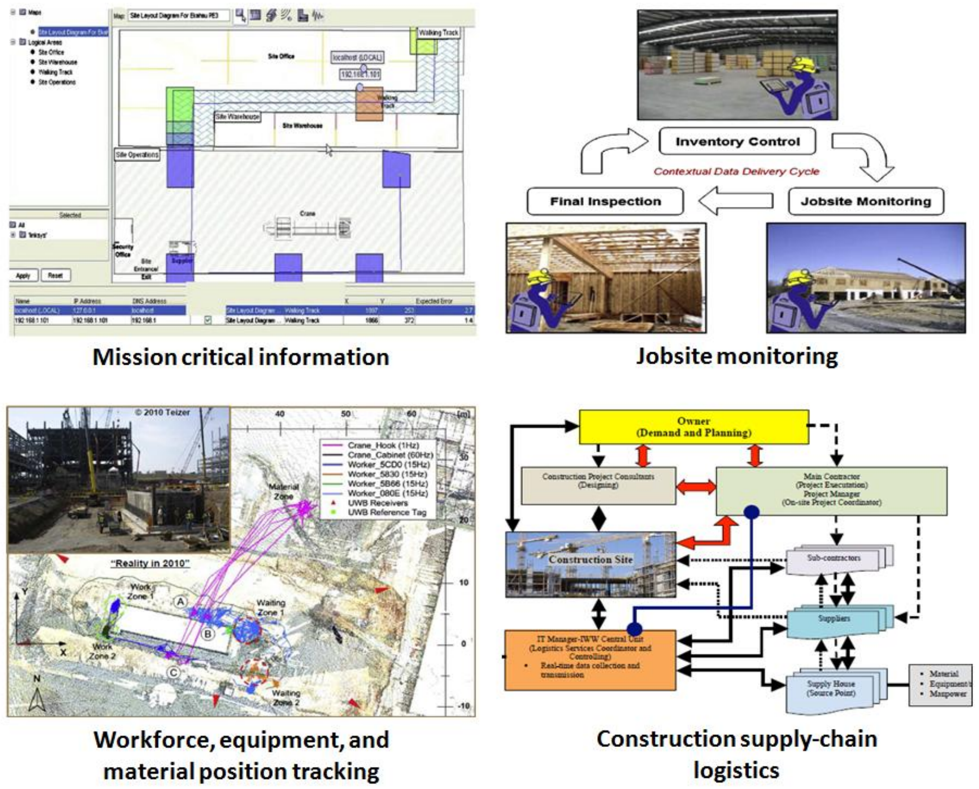


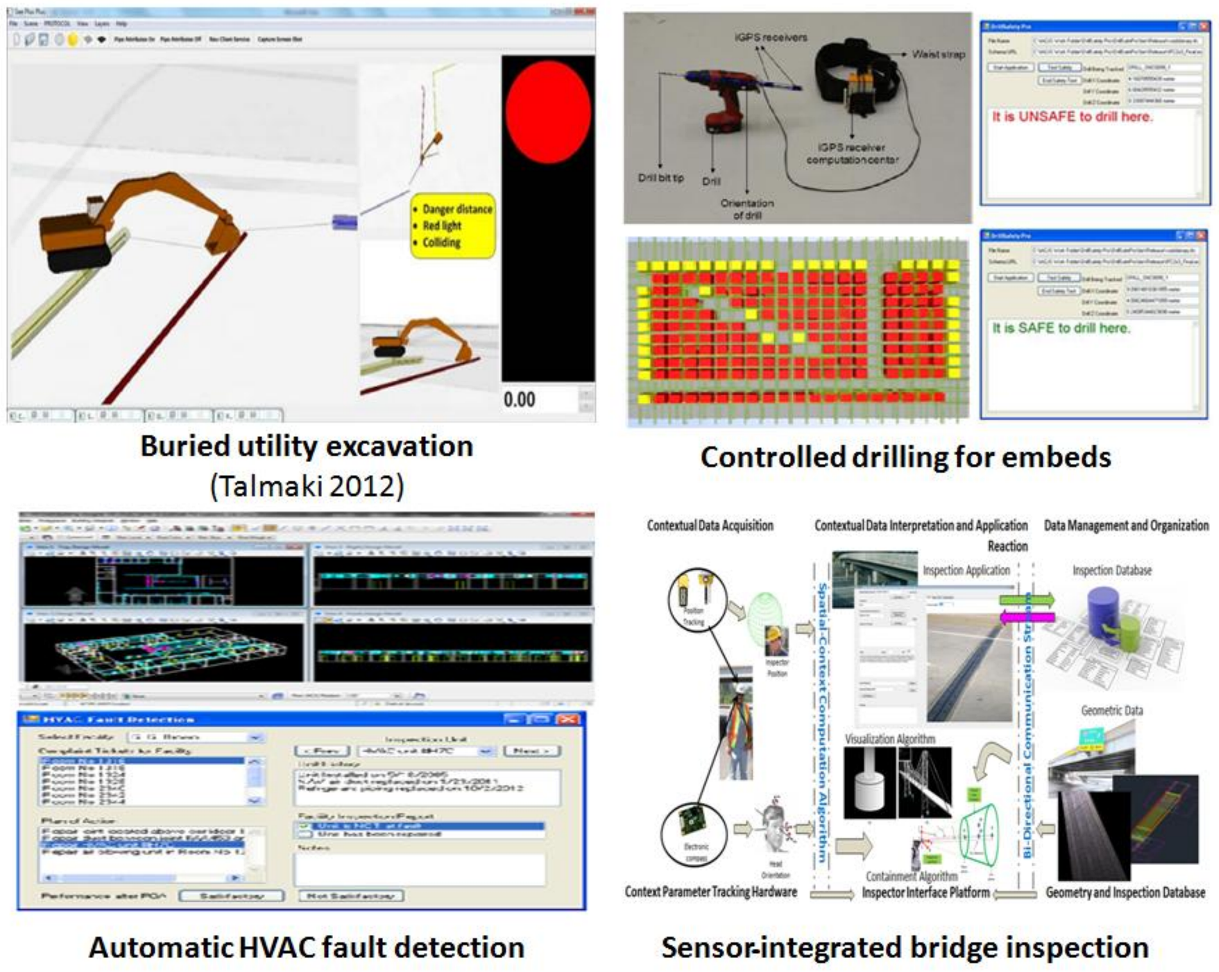
Figure 1.2: Examples of context-aware applications in the civil infrastructure industry (Aziz et al 2006; Behzedan et al 2008; Teizer et al 2008; Omar and Ballal 2009)

1.2.3 Context-Aware Decision-Making Research in LIVE

In the aforementioned efforts, the location of mobile agents and physical assets are determined with relatively low accuracy, and the high uncertainty in the tracked location reduces the granularity of the spatial-context being interpreted (Khoury 2009). Moreover, in these efforts, the spatial context of a mobile agent is defined solely by the location leading to an incomplete and inaccurate interpretation of the agent's spatial context.

A context-aware computing framework has been developed at the University of Michigan's Laboratory for Interactive Visualization in Engineering (LIVE) to incorporate a mobile human agent's three-dimensional head orientation to define spatial-context (Khoury and Kamat 2009b). The location and the three-dimensional head orientation determine the line of sight of the mobile user, and the human's spatial-context is computed as a truncated pyramid using the concept of a viewing frustum, borrowed from computer graphics literature. The framework's interpretation of a human agent's spatial-context leads to much greater precision than is possible with location alone. This framework is used to develop prototype applications to automatically retrieve and visualize contextual information on identified visible components and data objects in the studied areas.

LIVE has also been engaged in developing context-aware decision-making frameworks and applications related to bridge retrofitting, facility management, and civil infrastructure maintenance. These applications, shown in Figure 1.3, are as follows: 1) real-time collision avoidance for excavation operations; 2) a real-time controlled drilling application for embeds into reinforced concrete decks; 3) automated fault detection in Heating, Ventilation, and Air Conditioning (HVAC) systems in operational buildings; and 4) pervasive sensor-network–integrated bridge inspection.


Danger distance
Red light
Colliding
0.00
Buried utility excavation
(Talmaki 2012)
iGPS receivers
Waist strap
Drill bit tip
Drill
Orientation of drill
iGPS receiver computation center
Controlled drilling for embeds
HVAC Fault Detection
Automatic HVAC fault detection
Contextual Data Acquisition
Contextual Data Interpretation and Application Reaction
Data Management and Organization
Inspection Application
Inspection Database
Geometric Data
Visualization Algorithm
Containment Algorithm
Spatial-Context Computation Algorithm
Bi-Directional Communication Stream
Position Tracking
Inspector Position
Electronic compass
Head Orientation
Context Parameter Tracking Hardware
Inspector Interface Platform
Geometry and Inspection Database
Sensor-integrated bridge inspection

It is UNSAFE to drill here.

It is SAFE to drill here.

Figure 1.3: Context-aware applications developed in LIVE

The collision avoidance system for excavation operations determines, in real-time, what utilities lie buried in the vicinity of an excavator, and the system warns the operator when the excavator is in the vicinity of any utility line, thus preventing accidents caused by utility strikes (Talmaki 2012). The controlled drilling system determines, in real-time, whether the drilling equipment is about to strike any rebar or utility lines while drilling for embeds into reinforced concrete decks, thus preventing any compromise in the structural integrity of the deck, and thus avoiding accidents caused by utility strikes (Akula et al 2013a). In an HVAC system failure scenario, the automated fault detection application predicts the combination of HVAC components most likely at fault thus allowing maintenance personnel to streamline their operations (Golabchi et al 2013). Finally, the pervasive sensor-network–integrated bridge inspection system augments visual inspection methods with contextual sensor analytics and life cycle analysis methods to assess bridge component condition, thus allowing for a more objective and data-rich documentation of bridge inspection ratings and reports (Akula et al 2013b).

1.3 Detailed Research Challenges

1.3.1 Ubiquitous Localization

Spatial context is one of the most important parameters considered in context-aware computing. For context-aware engineering applications, the ability to ubiquitously track a mobile agent’s location continuously and accurately is of the utmost importance. Several infrastructure-based localization systems have been developed and are commercially available for outdoor (GPS) and indoor (WLAN, Ultra-Wide Band (UWB) and Indoor GPS based systems) localization. The main drawback of these technologies is their dependency on pre-installed infrastructure, which makes them unsustainable in a dynamic environment that cannot be prepared or retrofitted a priori.

In order to overcome this drawback, several infrastructure-independent positioning systems have been developed in recent years. Typically, infrastructure-independent positioning systems are based on inertial measurements and make use of high-accuracy gyroscopes and accelerometers. The main drawback of infrastructure-independent positioning systems based on inertial measurements is the accumulated drift error that grows with the distance travelled by the mobile agent (Akula et al 2011; Ojeda and Borenstein 2007; Davidson et al 2010).

Therefore, context-aware computing applications in civil engineering would greatly benefit from ubiquitous localization systems that seamlessly integrate infrastructure-dependent localization systems with infrastructure-independent localization technologies to continuously and effectively track the location and trajectory of mobile agents in dynamic environments. Such ubiquitous localization, when combined with technologies that track a mobile agent's angular orientation, would help determine the fully qualified spatial-context of the agent.

1.3.2 Interpreting Contextual Objects

The ability of a context-aware computing application to interpret the spatial contextual status of objects in its environment depends on the precision with which the geometry of the agent's spatial context can be constructed, which in turn depends on the accuracy of the tracking technology employed. Spatial context is often abstracted by constructing representative geometric models of the region of influence and the field of view for machine and human agents, respectively. These geometric models are derived from location and angular fundamentals, which are monitored ubiquitously using sensing technologies.

Error in the location and angular fundamentals of the mobile agent induces error in the constructed spatial context. A combination of errors in tracking technologies may add or cancel each other's effects while contributing to an error in the interpreted spatial context. While there is plenty of literature that characterizes the nature of errors in the aforementioned tracking technologies (Khoury and Kamat 2009b; Mautz 2009), it is difficult to predict the error in the location and angular fundamentals at a particular instance. If a combination of multiple tracking technologies is employed to determine the spatial-context, it may lead to further complications in interpreting spatial-context. It is therefore a primary concern to investigate the effects of the aforementioned errors on the interpretation of spatial context and provide suitable methodologies for a context-aware computing application to override such potential errors.

1.3.3 Data Management and Organization

In order to implement context-aware computing applications for contextual information access and retrieval for real-time decision-making in construction, inspection, and maintenance of civil infrastructure facilities, it is imperative to have a methodology to store the physical and functional characteristics of these facilities. Previously used standard product models and databases for civil engineering facility construction, inspection, and maintenance have included Microsoft Access Project databases and CIMSteel Integration Standards (CIS/2) product models (Khoury 2009). However, these databases are typically designed from scratch and are specialized for the application they are intended to address.

Building Information Modeling (BIM) is a powerful tool that can be used as the data model for context-aware applications in the building facilities inspection and management domain; however there are no counterparts to BIM to develop context-aware applications for the inspection and maintenance of civil infrastructure such as bridges. Bridge infrastructure maintenance has been of special interest in recent years with more than 31% of bridges in the United States being considered deficient. Therefore, it is imperative to develop a bridge information model that serves context-aware bridge maintenance applications.

While standard product models and databases such as CIS/2, BIM, and bridge information modeling system can be used to effectively store infrastructure data, they may consist of information that is considered superfluous for the application and the task at hand. For example, consider the use of BIM as the database for developing context-aware applications for HVAC facilities maintenance. The information in the BIM regarding facility systems that are not within the scope of the task (for example structural, electrical, and plumbing systems) is superfluous and may lead to additional computations while determining context, and will present the user with redundant data, thus making the applications computationally inefficient. In order to avoid these problems, the context-aware computing applications must accommodate methodologies to screen information based not just on spatial contextual parameters but also on functional contextual parameters such as user, task, and activity profiles.

1.4 Research Objectives

The objectives of the research were as follows:

- Investigate the architecture of currently available context-aware applications in the civil infrastructure and other related domains.
- Develop a strategic framework and identify key drivers in designing, developing, and implementing real-time context-aware decision-making applications for the civil engineering domain.

- Investigate currently existing technologies to track spatial contextual parameters and create a ubiquitous localization system for tracking human agents by integrating infrastructure-dependent and inertial navigation technologies.
- Characterize and evaluate the effect of errors in position-tracking technologies on uncertainties in constructing representative geometric models for interpreting the spatial context of mobile agents.
- Investigate the potential of adopting standard product models and databases such as CIS/2 and BIM to support context-aware applications for operational buildings, and create a bridge information modeling system to support context-aware applications for bridge infrastructure facilities.
- Based on the developed strategic framework, design and validate methods for real-time monitoring to control drilling for embeds into reinforced concrete decks to avoid striking rebar and buried utility lines.
- Extend the developed framework to implement automated fault detection methods—to support repair operations of a failed HVAC system—that predict the HVAC components most likely responsible for the deteriorated performance, by leveraging environmental and activity status contextual parameters.
- Based on the developed strategic framework, design a scalable, extensible context-aware bridge inspection application to query, access, retrieve, and document digital and multimedia condition assessment information from multiple sources such as visual assessment, pervasive sensor network analytics, and lifecycle analysis methods.

The end result of pursuing these research objectives is a strategic framework for developing context-aware applications in the civil infrastructure domain and an application suite. The application suite contains the following applications: 1) Integrated Tracking System, 2) Controlled Drill Monitoring System, 3) HVAC Fault Detection System, and 4) Bridge Inspection System. These applications are designed as a combination of multiple drivers and modules that provide the context-awareness and functionality required to address the specific scenarios presented in this dissertation.

1.5 Research Methodology

The methodology is comprised of different thrusts, each having a specific focus. The overview of the research methodology and its key components are shown in Figure 1.4.

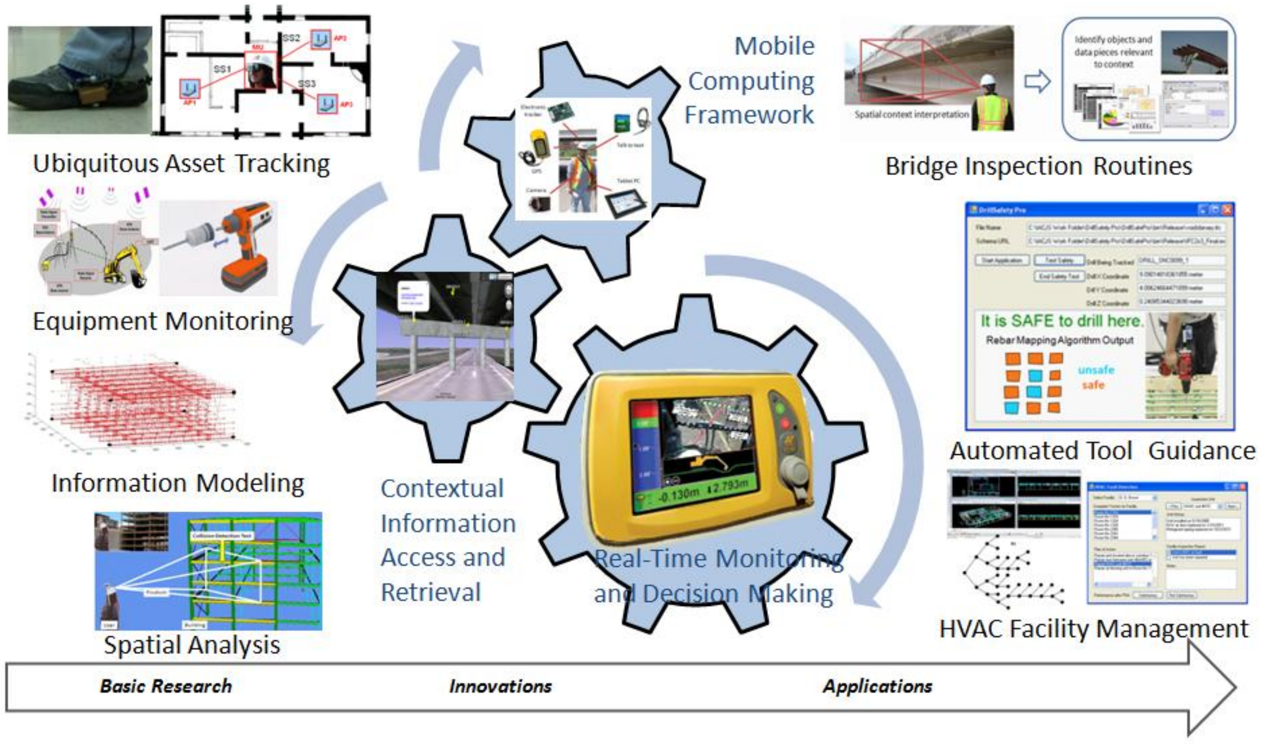

Ubiquitous Asset Tracking
Equipment Monitoring
Information Modeling
Spatial Analysis
Mobile Computing Framework
Contextual Information Access and Retrieval
Real-Time Monitoring and Decision Making
Bridge Inspection Routines
Identify objects and data pieces relevant to context
Spatial context interpretation
Automated Tool Guidance
It is SAFE to drill here.
Rebar Mapping Algorithm Output
unsafe
safe
HVAC Facility Management
Basic Research
Innovations
Applications

Figure 1.4: Overview and primary thrusts of the research methodology

- Developed a framework for designing real-time context-aware applications in the civil infrastructure and related industries.
    - Identified key drivers in designing, developing, and implementing the architecture for real-time context-aware applications.
    - Classified commonly used contextual parameters and identified existing strategies and methodologies to sense and acquire these parameters.
    - Identified the requirements and goals for designing data models suitable for context-aware applications. Identified existing standard product models and databases to serve this purpose.
    - Identified methods to construct and interpret context for human and machine agents in order to make and communicate decisions in real-time.
- Designed a ubiquitous localization algorithm that integrated infrastructure-based localization systems with inertial navigation systems to continuously track a mobile human agent.
    - Developed an integrated tracking system by integrating personal dead reckoning navigation with GPS for ubiquitous localization.
    - Developed an integrated tracking system by integrating personal dead reckoning navigation with location marker database for ubiquitous localization.
    - Developed a human-intelligence–based tracking system by integrating personal dead reckoning navigation with a human's perception of their environment.
    - Validated the integrated tracking systems in dynamic environments simulating civil infrastructure worksites.

- Developed a method for controlled drilling for embeds into reinforced concrete decks to warn drilling personnel, in real-time, when they are about to strike rebar or buried utilities.
    - Reviewed methods currently employed to drill for embeds into reinforced concrete decks, and the shortcomings of these methods.
    - Investigated methods to document and visualize rebar locations and the geometry of zones that are safe/unsafe for drilling based on processing point cloud data of rebar cages.
    - Developed an algorithm that tracks the position of the drill bit tip and warns the drilling personnel when they are about to strike rebar or utility lines.
    - Validated the controlled drilling system by evaluating the accuracy of the position-tracking system used to localize the drills and the accuracy of the rebar mapping and safe/unsafe zone prediction algorithms used in the system.
- Developed an automated fault detection method for assisting field inspectors and repair personnel in developing a plan of action to address complaints regarding an underperforming/faulty HVAC system.
    - Modeled the deterioration in HVAC performance based on the complaint tickets generated by the occupants of operational buildings.
    - Leveraged BIM models of the HVAC distribution network system to predict the HVAC components that are most likely at fault and are thus responsible for the deterioration in HVAC performance.

    - Developed an algorithm that dynamically generates a plan of action for the field inspectors and repair personnel to follow based on contextual parameters such as sensed HVAC system performance and inspection status feedback.
- Integrated existing bridge inspection methods with a context-aware computing framework and sensor analytics methods to design bridge inspection software to augment and support bridge inspection routines.
    - Developed a context-aware computing framework to eliminate the time and effort associated with manually navigating bridge inspection software by automatically identifying the bridge components that are of interest to the inspector.
    - Designed and implemented a relational database for storing physical, functional characteristics of bridge infrastructure, and inspection records.
    - Develop a scalable bridge inspection application that queries, accesses, retrieves, and documents digital and multimedia information about bridge component condition assessment across infrastructure fleets.
    - Extended the bridge inspection application to allow the inspector to access bridge component condition assessment from multiple sources, such as traditional visual assessment, pervasive sensor-network-analytics–based assessment, and lifecycle analysis methods.

The research mainly focused on establishing the developed context-aware computing framework and demonstrating its robustness in addressing domain specific problems in the civil infrastructure industry. To this end, the research addresses the domain specific problems by implementing solutions based on the designed framework. The implemented solutions comprise of compartmentalized methods that are developed using currently available algorithms and technologies, and that can be easily replaced by suitable alternatives. For example, consider the case of automating the fault detection process for assisting field personnel to repair HVAC systems. The research leverages an existing fault detection algorithm (Leckie and Dale 1997) and does not delve on optimizing the algorithm for HVAC fault detection scenario, purposely. However, the developed solution allows the adopted algorithm to be replaced with better algorithms—that identify faults more accurately—if developed in the near future.

1.6 Dissertation Outline

This dissertation is a compilation of peer-reviewed scientific manuscripts that document the research involved in designing the strategic framework for developing context-aware civil infrastructure applications and the corresponding derived applications. Chapter 2 describes the key components of the framework and the approach developed for designing context-aware applications. Chapters 3 through 7 are stand-alone papers that describe details of basic research questions answered, scientific issues investigated, scenarios and challenges encountered, alternatives considered, methods adopted, innovations developed, and applications implemented in addressing a specific research issue or a set of related issues.

The dissertation concludes with chapter 8, which details the contributions and achievements of this research and presents directions for future work. All chapters have been written such that they can be easily read, clearly understood, and successfully replicated by a technically literate audience from diverse domains including those without any prior knowledge or experience in context-aware computing and/or real-time application development and programming. Since the chapters are written as self-contained papers, some information appears in multiple chapters for the sake of completeness. The outline of this dissertation for the remainder of the chapters, their titles, and a brief introduction are discussed below:

- **Chapter 2 – Integrating Context-Aware Computing with Real-Time Decision-Making**

This chapter focuses on developing a strategic framework and a structured approach toward developing context-aware applications for supporting real-time decision-making in the civil infrastructure domain. The main questions addressed in this chapter are as follows: 1) What are the criteria and standards to classify systems as real-time? 2) What are the key components and drivers in developing context-aware applications, and what are their inter-dependencies? 3) What are the commonly encountered contextual parameters in the domain that can be acquired to support real-time monitoring and decision-making applications? 4) How can one construct and interpret context for human and machine agents and how can one leverage the constructed context in making and communicating decisions to the end user agents?

First, a classification of real-time systems and their deadlines—based on the nature and strictness of their deadlines—is presented. Second, a review of currently existing approaches, middle ware, context models, and design principles for supporting adaptive mobile context-aware applications is carried out. Third, a framework for designing context-aware applications in the civil infrastructure domain is created. The framework acquires contextual parameters and relevant facility data to construct the context, interprets the context, makes decisions based on pre-defined contextual rules, and communicates the decisions to the end user agent. A classification of contextual parameters that are typically used in the civil infrastructure domain is then presented. Next, the criteria for designing data models for context-aware applications are presented and standard product models are evaluated based on these criteria. Methods to construct the context based on the sensed contextual parameters for human and machine agents are then presented. Finally, methods to interpret context, make and communicate decisions to the end user agent are discussed.

- **Chapter 3 – Ubiquitous Localization for Monitoring Spatial Context**

This chapter focuses on developing ubiquitous localization methods to continuously and accurately track a mobile human agent by integrating infrastructure-based localization technologies with infrastructure-independent localization technologies. In particular, the main questions addressed in this chapter are as follows: 1) What are the currently existing localization technologies used to track mobile human agents? 2) What are the uncertainties and shortcomings associated with these localization technologies? 3) How can we leverage inertial tracking systems to substitute for infrastructure-based localization systems when infrastructure-based localization is either unavailable or partially/fully damaged?

First, a thorough review of existing infrastructure based and infrastructure independent tracking technologies is carried out. Second, a technical approach toward integrating infrastructure-based localization technologies with inertial navigation technologies is presented. Third, three integrated tracking systems are designed, implemented, and validated based on the proposed technical approach. The first integrated tracking system developed integrates GPS localization technology with a Personal Dead Reckoning (PDR) navigation system to track human agents ubiquitously in a dynamic environment that typifies a civil infrastructure facility. The second integrated tracking system developed integrated the PDR navigation system with a database of navigation markers—whose location is known apriori—for ubiquitous localization. Finally, a hybrid tracking system that integrates the PDR navigation system with a human agent's ability to perceive and comprehend its environment is presented.

- **Chapter 4 – Controlled Drilling for Embeds into Reinforced Concrete Decks**

This chapter focuses on developing methods for controlled drilling for embeds into reinforced concrete decks to prevent damage to life and property caused by striking rebar and buried utility lines. The main questions addressed in this chapter are as follows: 1) what are the current methods used to drill for embeds into reinforced concrete decks? 2) How can we acquire contextual parameters to construct the spatial context of the drilling equipment? 3) How do we acquire and store information about the physical and functional characteristics of reinforced concrete decks? 4) How can we leverage context-awareness to monitor and control drilling for embeds? 5) What are the challenges involved in adopting context-aware monitoring methods developed for controlled drilling for embeds into the construction field environment?

First, a brief review of accidents and fatalities in the civil infrastructure industry is carried out, and the importance and necessity for hazard recognition is discussed. Hazard recognition methods are required to avoid striking rebar and buried utilities while drilling for embeds into reinforced concrete slabs and decks. Second, a thorough review of currently employed methods for safely drilling for embeds into reinforced concrete decks and their shortcomings is carried out. Third, a technical approach for context-aware controlled drilling for embeds is presented.

Methods to acquire, interpret, and store information regarding zones that are safe/unsafe for drilling are discussed in detail. Four, methods to track drilling equipment and laser projectors—which are used to guide drilling personnel—and corresponding algorithms that give feedback to the drilling personnel to help avoid striking rebar and buried utility lines are presented. The proposed methods are validated by evaluating the uncertainties in the methods adopted in this study. Finally, the limitations of the research study and the challenges in transferring the developed technology into the field are then presented.

- **Chapter 5 – Automated Fault Detection for HVAC Systems in Operational Buildings**

This chapter focuses on developing methods to repair deteriorated HVAC systems in operational buildings by automatically identifying HVAC components that are most likely at fault by automatically generating a plan of action for the repair personnel to follow. In particular, this chapter addresses the following questions: 1) what are the application domains where BIM can be used as a data model? 2) What are the technological approaches possible for leveraging BIM and context-awareness for fault detection in operational buildings in the case of HVAC system failure? 3) What are the available algorithms to detect faults in HVAC distribution network trees based on the deterioration in HVAC performance as reported by the building occupants?

First, a brief introduction to facility management in operational buildings is presented. Second, a thorough review of BIM and its potential applications as a data model is carried out. Third, the motivating scenario of HVAC system failure is presented. BIM is proposed to be used as the data model for developing a context-aware application that provides decision-making support to

HVAC repair personnel. Fourth, methods to map HVAC distribution network systems and to model the deterioration in HVAC performance based on the complaint tickets reported by occupants are then presented. A fault detection algorithm from network and computer science literature is adopted to create a context-aware HVAC system repair workflow methodology. Finally, the advantages and challenges of utilizing automated context-aware fault detection methods to generate repair plans for field personnel are discussed.

- **Chapter 6 – Context-Aware Framework for Highway Bridge Inspections**

This chapter focuses on developing methods to automatically navigate bridge inspection software by identifying the bridge components that are of interest to the mobile inspector by ubiquitously sensing and interpreting the inspector’s context. The main questions this chapter addresses are as follows: 1) how do we integrate context-awareness into bridge inspection software to eliminate the time and effort associated with the manual navigation of the software? 2) What are the technologies and methods involved in tracking the mobile inspector’s contextual parameters to construct the context? 3) How can we predict the bridge component of interest to the inspector based on the constructed context? 4) What is the run-time and space complexity of the algorithms used to predict the component of interest? 5) How do the uncertainties in tracking technologies employed affect the accuracy of the prediction performance of the application? 6) What are the challenges and limitations of integrating context-awareness with bridge inspection software for conducting inspection routines in the field?

First, a brief introduction to highway bridge inspections and the software technology currently used in damage condition assessment is presented. Second, a context-aware computing application is presented in order to automatically navigate the bridge inspection software based on the sensed context. Third, methods to track the inspector's contextual parameters, construct the inspector's context, identify the bridge components in context, and predict the component that is most likely of interest are discussed. Fourth, the context-aware application is evaluated based on the run-time and space complexity of the containment and prioritization algorithms used to predict the component of interest. Methods to evaluate the application's prediction performance using simulated sensitivity analysis and field validation experiments are then presented. Finally, the results of the sensitivity analysis and the field validation experiments are presented and the limitations of adopting this technology in the field are discussed.

- **Chapter 7 – Context-Aware Bridge Inspection Routines**

This chapter focuses on the efforts made toward designing, developing, and implementing context-aware bridge inspection routines by leveraging pervasive condition assessment sensor network technology to support and augment traditional visual damage condition assessment methods. In particular, the main questions addressed in this chapter are as follows: 1) what is the importance of bridge condition assessment and how can we integrate context-awareness into bridge inspection routines? 2) How can we effectively store information corresponding to the physical, functional, and condition assessment characteristics of bridge infrastructure? 3) What technical methods can be adopted to leverage context-aware pervasive condition assessment network and lifecycle analysis methods to support bridge inspectors in their routines? 4) What are the challenges and implications of implementing context-aware bridge inspection methods in bridge inspection routines?

First, a brief introduction to the importance and overview of bridge condition assessment in maintaining an ageing fleet of bridge infrastructure is presented along with a review of the methods to integrate context-awareness into bridge condition assessment routines. Second, a relational bridge data model designed to store physical, functional, and bridge condition assessment/inspection information is presented. Third, the scalable and extensible bridge inspection software developed as a part of this research to query, access, retrieve, and document digital and multimedia condition assessment information is presented and its functionality is discussed. Fourth, the software is extended to incorporate multiple damage condition assessments methods such as traditional visual assessment, and contextual sensor-analytics–based assessment methods. A method to use contextual analytics of a pervasive condition assessment sensor network to guide bridge inspection routines is then presented. Finally, the advantages of using context-aware bridge inspection methods over traditional inspection methods, the challenges encountered, and the future work required for successfully implementing the developed methods in the field are discussed.

- **Chapter 8 – Conclusion**

This chapter summarizes the contributions and achievements of the research presented in this dissertation. The significance of this research in achieving the desired objectives and solving the proposed problems is discussed in detail. The chapter concludes the dissertation by suggesting potential directions for future research.

1.7 References

Abowd, G.D., Atkeson, C.G., Hong, J., Long, S., Kooper, R., and Pinkerton, M. (1997). "Cyberguide: A Mobile Context-aware Tour Guide," Wireless Networks – Special Issue: Mobile Computing and networking, Vol. 3(5), pp. 421–433.

Aittola, M., Ryhanen, T., and Ojala, T. (2003). "SmartLibrary – Location-Aware Mobile Library Service," Proc. of the 5th Int. Symp. on Human Computer Interaction with Mobile Devices and Services, Udine, Italy, pp. 411–416.

Akula, M., Dong, S., Kamat, V.R., Ojeda, L., Borrell, A., and Borenstein, J. (2011). "Integration of Infrastructure Based Positioning Systems and Inertial Navigation for Ubiquitous Context-Aware Engineering Applications," Advanced Engineering Informatics, Vol. 25(4), ICCCBE 2010 Special Issue on Computing in Civil and Building Engineering, Elsevier Science, New York, NY, pp. 640–655.

Akula, M., Lipman, R.R., Franaszek, M., Saidi, K.S., Cheok, G.S., and Kamat, V.R. (2013a). "Real-Time Monitoring: Augmenting Building Information Modeling with 3D Imaging Data to Control Drilling for Embeds into Reinforced Concrete Bridge Decks," Automation in Construction (In Review).

Akula, M., Sandur, A., Kamat, V.R., and Prakash, A. (2013b). "Evaluation of Context-Aware Computing for Improved Bridge Inspections," Journal of Computing in Civil Engineering, American Society of Civil Engineers, Reston, VA. (In Press)

Asthana, A., Cravatts, M., and Krzyzanowski, P. (1994). “An Indoor Wireless System for Personalized Shopping Assistance,” Proc. of IEEE Workshop on Mobile Computing Systems and Applications, IEEE Computer Society Press, Santa Cruz, California, pp. 69–74.

Aziz, Z., Anumba, C.J., Ruikar, D., Carrillo, P.M., and Bouchlaghem, D.N. (2005). “Context Aware Information Delivery for on-Site Construction Operations,” 22nd CIBW78 Conference on Information Technology in Construction, Institute for Construction Informatics, Technische Universitat Dresden, Germany, CBI Publication No:304, pp. 321–332.

Aziz, Z., Anumba, C.J., Ruikar, D., Carrillo, P.M., and Bouchlaghem, D.N. (2006). “Intelligent Wireless Web Services for Construction – A review of Enabling Technologies,” Automation in Construction, Vol. 15(2), Elsevier, 113–123.

Bachmann, T., Naab, K., Reichart, G., and Schraut, M. (2000). “Enhancing Traffic Safety With BMW’s Driver Assistance Approach Connected Drive,” Proc. of the 7th World Congress on Intelligent Transportation Systems Conference, Turin, Italy, pp. 21–24.

Bahrami, A., Yuan, J., Smart, P.R., and Shadbolt, N.R. (2007). “Context-Aware Information Retrieval for Enhanced Situation Awareness,” Proc. of the 2007 Military Communications Conference (MILCOM’07), Institute of Electrical and Electronics Engineers (IEEE), Orlando, Florida, pp. 1–6.

Bardram, J.E. (2004). “Applications of Context-Aware Computing in Hospital Work – Examples and Design Principles,” Proc. of the 2004 ACM Symposium on Applied Computing, Association for Computing Machinery (ACM), Nicosia, Cyprus, pp. 1574–1579.

Behzadan A. H., Aziz Z., Anumba C., and Kamat V. R. (2008). “Ubiquitous Location Tracking for Context Specific Information Delivery on Construction Sites,” Automation in Construction, Vol. 17(6), Elsevier Science, New York, NY, pp. 737–748.

Beigl, M. (2000). “MemoClip: A Location-based Remembrance Appliance,” Personal Technologies, Vol. 4(4), Springer, pp. 230–233.

Black, D., Clemmensen, N.J., and Skov, M.B. (2009). “Supporting the Supermarket Shopping Experience through a Context-aware Shopping Trolley,” Proc. of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7, Melbourne, Australia, pp. 33–40.

Brown, P.J., Bovey, J.D., and Chen, X. (1997). “Context-aware Applications: From the Laboratory to the Marketplace,” IEEE Personal Communications, Institute of Electrical and Electronics Engineers (IEEE), Vol. 4(5), pp. 58–64.

Burrell, J., and Gay, K. (2001). “Collectively Defining Context in a Mobile, Networked Computing Environment,” Proc. of the Conference on Human Factors in Computing Systems, Association for Computing Machinery (ACM), New York, NY, pp. 231–232.

Chen, G., and Kotz, D. (2000). "A Survey of Context-aware Mobile Computing Research," Dartmouth Computer Science Technical Report, TR2000-381, Department of Computer Science, Dartmouth College, Hanover, NH, pp. 1–16.

Cheverst, K., Davies, N., Mitchell, K., Friday, A., and Efstratiou, C. (2000). "Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences," Proc. of the 2000 SIGCHI Conference on Human Factors in Computing Systems, Association for Computing Machinery (ACM), The Hauge, Netherlands, pp. 17–24.

Davidson, P., Collin, J., and Takala, J. (2010). "Application of Particle Filters for Indoor Positioning using Floor Plans," IEEE Conference on Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS 2010), Institute of Electrical and Electronics Engineers (IEEE), Kirkkonummi, Finland, pp. 1–4.

Davies, N., Cheverst, K., Mitchell, K., and Friday, A. (1999). "Caches in the Air: Disseminating Information in the Guide System," Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), Institute of Electrical and Electronics Engineers (IEEE), New Orleans, LA, pp. 11–19.

Dey, A.K., and Abowd, G.D., (2000). "CybreMinder: A Context-aware System for Supporting Reminders," Proc. of 2nd Int. Symp. on Handheld and Ubiquitous Computing (HUC 2000), Bristol, UK, pp. 172–186.

Fathi, M.S., Anumba, C.J., and Carrillo, P. (2006). "Context Awareness in Construction Management – Key Issues and Enabling Technologies," Proc. of the Joint Int. Conference on Construction Culture, Innovation, and Management (CCIM'06), Dubai, UAE, pp. 425–435.

Fischmeister, S., Menkhaus, G., and Pree, W. (2002). "MUSA-Shadow: Concepts, Implementation, and Sample Applications," Proc. of the 40th Int. Conference on Tools Pacific, Australian Computer Society (ACS), Sydney, Australia, pp. 71–79.

Fleck, M., Frid, M., Kindberg, T., O'Brien-Strain, E., Rajani, R. and Spasojevic, M. (2002). "From Informing to Remembering: Deploying Ubiquitous Systems in Interactive Museums," IEEE Pervasive Computing, Vol. 1(2), pp. 13–21.

Gizmodo (2013). "Augmented Reality Yelp Will Murder All Other iPhone Restaurant Apps, My Health" <http://gizmodo.com/5347194/> as accessed on 28th March, 2013.

Golabchi, A., Akula, M., and Kamat, V.R. (2013). "Leveraging BIM for Automated Fault Detection in Operational Buildings," The 30th Int. Symp. on Automation and Robotics in Construction and Mining (ISARC 2013), Montreal, Canada (In Press).

Hoch, S., Schweigert, M., Althoff, F., and Rigoll, G. (2007). "The BMW SURF Project: A Contribution to the Research on Cognitive Vehicles," Proc. of the 2007 IEEE Vehicles Symposium, Istanbul, Turkey, pp. 692–697.

Huang, A.C., Ling, B.C., Ponnekanti, S., and Fox, A. (1999). “Pervasive Computing: What is it good for?” Proc. of the ACM International Workshop on Data Engineering for Wireless and Mobile Access, Association for Computing Machinery (ACM), Seattle, WA, pp. 84–91.

Jahnke, J.H., Bychkov, Y., Dahlem, D., and Kawasme, L. (2005). “Context-Aware Information Delivery in Health Care,” Revue d’Intelligence Artificielle, Vol. 19(3), pp. 459–478.

Khoury, H. M. (2009). "Context Aware Information Access and Retrieval for Rapid On-Site Decision Making in Construction, Inspection and Maintenance of Constructed Facilities,” Ph.D. Dissertation, Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI.

Khoury, H. M., and Kamat, V. R. (2009a). "High-Precision Identification of Contextual Information in Location-Aware Engineering Applications,” Advanced Engineering Informatics, Vol. 23 (4), Elsevier Science, New York, NY, pp. 483–496.

Khoury, H. M., and Kamat V. R. (2009). "Indoor User Localization for Context-Aware Information Retrieval in Construction Projects,” Automation in Construction, Vol. 18(4), Elsevier Science, New York, NY, pp. 444–457.

Kortuem, G., Bauer, M., and Segall, Z. (1999). “NETMAN: the Design of a Collaborative Wearable Computer System,” Mobile Networks and Applications, Vol. 4(1), pp. 49–58.

Laukkanen, M., Helin, H., and Laamanen, H. (2002). “Tourists on the Move,” Cooperative Information Agents VI, Springer Berlin Heidelberg, Germany, pp. 36–50.

Leckie, C. and Dale, M. (1997). “Locating Faults in Tree-Structured Networks,” Proc. of the 15th Int. Joint Conference on Artificial Intelligence (IJCAI), Nagoya, Japan, pp. 434–439.

Mautz, R. (2009). “Overview of Current Indoor Positioning Systems,” Geodezija Ir Kartografija / Geodesy and Cartography, Vol. 35(1), pp. 18–22.

Marmasse, N., and Schmandt, C. (2000). “Location-aware Information Delivery with ComMotion,” Proc. of 2nd Int. Symp. on Handheld and Ubiquitous Computing, (HUC 2000), Bristol, UK, pp. 157–171.

Ojeda, L., and Borenstein, J. (2007). “Personal Dead Reckoning System for GPS-denied Environments,” IEEE International Workshop on Safety, Security and Rescue Robotics, Institute of Electrical and Electronics Engineers (IEEE), Rome, Italy, pp. 1–6.

Omar, B., and Ballal, T. (2009). “Intelligent Wireless Web Services: Context-aware Computing in Construction-logistics Supply Chain,” Journal of Information Technology in Construction (ITcon), Vol. 14, pp. 289–308.

Ordonez, P., Kodeswaran, P., Korolev, V., Li, W., Walavalkar, O., Elgamil, B., Joshi, A., Finin, T., Yesha, Y., and George, I. (2007). “A Ubiquitous Context-aware Environment for Surgical Training,” 4th Annual Conference on Mobile and Ubiquitous Systems: Networking & Services 2007 (MobiQuitous 2007), Institute of Electrical and Electronics Engineers (IEEE), Philadelphia, PA, pp. 1–6.

Pascoe, J., Morse, D.R., and Ryan, N.S. (1998). “Developing Personal Technology for the Field,” Journal of Personal and Ubiquitous Computing, Vol. 2(1), Springer, London, pp. 28–36.

Schilit, B.N., Adams, N., and Want R. (1994). “Context-aware Computing Applications,” Workshop on Mobile Computing Systems and Applications (WMCSA), Santa Cruz, CA, pp. 85–90.

Schwinger, W., Grun, C., Proll, B., and Retschitzegger, W. (2009). “Context-Awareness in Mobile Tourist Guides,” Handbook of Research on Mobile Multimedia, 2nd Edition, IGI Global, Hershey, PA, pp. 534–552.

Skibniewski, M.J., and Jang, W.S. (2006). “Ubiquitous Computing: Object Tracking and Monitoring in Construction Processes Utilizing ZigBee Networks,” Proc. of the 23rd Int. Symp. on Automation and Robotics in Construction (ISARC), Tokyo, Japan, pp. 287–292.

SleepCycle 3.0 (2013). “Sleep Cycle Alarm Clock,” < http://www.sleepcycle.com/> as accessed on 28th March, 2013.

Son, H., Kim, Ch., Kim, H., Choi, K.-N., and Jee, J.-M. (2008). “Real-time Object Recognition and Modeling for Heavy-Equipment Operation,” Proc. 25th Int. Symp. on Automation and Robotics in Construction, Vilnius, pp. 232–237.

Talmaki, S. (2012). "Real-Time Visualization for Prevention of Excavation Related Utility Strikes,” Ph.D. Dissertation, Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI.

Teizer, J.D., Venugopal, M., and Walia, A. (2008). “Ultra Wideband for Automated Real-time Three-Dimensional Location Sensing for Workforce, Equipment and Material Positioning and Tracking,” Proc. of the 87th Transportation Research Board Annual Meeting, Washington, D.C.

Thomas, B., Close, B., Donoghue, J., Squires, J., Bondi, P. D., Morris, M., and Piekarski, W. (2000). "ARQuake: An Outdoor/Indoor Augmented Reality First Person Application,” Proc. of the 2000 International Symposium on Wearable Computers, Atlanta, GA, pp. 75–86.

Want, R., Hopper, A., Falcao, V., and Gibbons, J. (1992). “The Active Badge Location System,” ACM Transactions on Information Systems, Association for Computing Machinery (ACM), Vol. 10(1), pp. 91–102.

Wen, Z., Zhou, M., and Aggarwal, V. (2007). “Context-aware, Adaptive Information Retrieval for Investigative Tasks,” Proc. of the 12th Int. Conference on Intelligent User Interfaces (IUI’07), Honolulu, Hawaii, pp. 121–131.

Yan, H., and Selker, T. (2000). “Context-aware Office Assistant,” Proc. of the 2000 Int. Conference on Intelligent User Interfaces (IUI’00), New Orleans, LA, pp. 276–279.

Zhu, W., Owen, C.B., Li, H., and Lee, J.H. (2004). “Personalized In-store e-Commerce with the Promopad: An Augmented Reality Shopping Assistant,” Electronic Journal for E-commerce Tools and Applications, Vol. 1(3), pp. 1–19.

Zou, W., Ye, X., and Chen, Z. (2006). “A brief Review on Application of Mobile Computing in Construction,” 1st Int. Multi-Symp. on Computer and Computational Sciences (IMSCCS’06), Vol. 2, Zhejiang University, Hangzhou, China, pp. 657–661.

Chapter 2

Real-Time Context-Aware Applications

2.1 Real-Time Systems

By their nature, projects in the civil infrastructure domain consist of mission critical tasks whose failure will result in the failure of operations. In order to avoid operational failure, it is highly desirable to have decision-making support in real-time or near real-time. Real-time systems are defined as systems whose operational effectiveness depends on both the logical correctness and the performance time. Real-time systems and their performance time deadlines are classified based on the consequences of failing to meet their deadlines, as shown below (Khaitan and Gupta 2013).

1) Hard real-time: Systems where missing a deadline leads to a total system failure.
2) Firm real-time: Systems where missing occasional deadlines is tolerable but where the usefulness of a result is zero after its deadline.
3) Soft real-time: Systems where the usefulness of a result degrades after its deadline thus degrading the quality of service.

A system is required to adhere to hard real-time standards if the operation it supports requires events to occur within strict deadlines. Such strong standards are usually required of systems for which not reacting within a certain deadline would result in a great loss to life or property. For example, consider a drill control system used to warn drill operators when they are about to strike rebar or utility lines while drilling for embeds into reinforced concrete decks (Akula et al 2013). Striking rebar might lead to a loss of structural integrity, and striking utility lines might result in injuries, casualties, disruption in service, and loss of property. The drill control must be designed such that the time required for the operator to react to the warning (or for the drill to shut down) is less than the time period after which the drill is expected to strike rebar or utility lines. Another example is proximity monitoring for prevention of excavation-related utility strikes to avoid the loss of life and property resulting from such accidents (Talmaki 2012). The excavator monitoring system warns the operator early enough to react and avoid striking the buried utility lines. It is integral that the drill control and excavator monitoring systems are designed with adherence to hard real-time standards, as failure to meet deadlines would result in a total system failure.

However, systems that can be allowed to operate with a reasonable latency are not required to adhere to hard real-time standards. For example, consider bridge inspection software systems that allow the inspector ubiquitous access to retrieve and document bridge condition assessment information (Adams et al. 2005; BridgeInspect^{TM} 2012; BridgeWorks.NET 2012; Indiana State BIAS 2012). The documentation and retrieval of bridge condition assessment information can be allowed to occur with a latency of a few seconds. However, if the latency increases to the order of minutes or more, then the inspection system would experience degraded performance. Another example is the automated HVAC fault detection system that, upon receiving a complaint of HVAC system failure, generates a plan of action for the HVAC repair personnel (Golabchi et al 2013). The automated HVAC fault detection system was designed to eliminate the time required to manually generate a plan of action based on the complaint tickets, HVAC system architecture, and the repair personnel's experience. Therefore, while a latency of a few seconds is acceptable, longer latencies result in a degraded performance as they undermine the purpose of automation. Systems such as the aforementioned bridge inspection software (Adams et al. 2005; BridgeInspect^{TM} 2012; BridgeWorks.NET 2012; Indiana State BIAS 2012) and the automated HVAC fault detection (Golabchi et al 2013) are considered soft real-time systems.

Research in the context-aware computing domain has largely focused on architectural approaches for supporting adaptive mobile context-aware applications. Researchers have identified several design principles and context models to design middleware and server-based approaches to ease the development of context-aware applications (Baldouf et al 2007). Frameworks (Beigel and Cahill 2004) and toolkits (Dey et al 2001; Efstratiou et al 2001) have been designed to ease the development of context-aware applications by allowing developers to fuse data from disparate sensors, represent contextual abstractions, and reason efficiently about context.

2.2 Architecture of Context-Aware Systems

The principles for developing context-aware computing applications remain largely similar across existing systems and frameworks. However, existing systems and frameworks use their own mechanisms to describe, interpret, reason, and communicate about context depending on the operation and domain being supported (Baldouf et al 2007). The remainder of this chapter presents a framework developed in this research for designing context-aware applications in the civil infrastructure domain, and discusses the key drivers involved in creating such applications.

The key components of the conceptual framework for designing and developing context-aware applications in the civil infrastructure domain are shown in Figure 2.1. The framework continuously and ubiquitously acquires the contextual parameters and has ubiquitous access to the operational facility data. The sensed contextual parameters and facility data are reconciled and the application's context is constructed. Pre-determined contextual rules that govern the operation are used to interpret the constructed context and make appropriate decisions that are then communicated using audio-visual techniques to the end user.

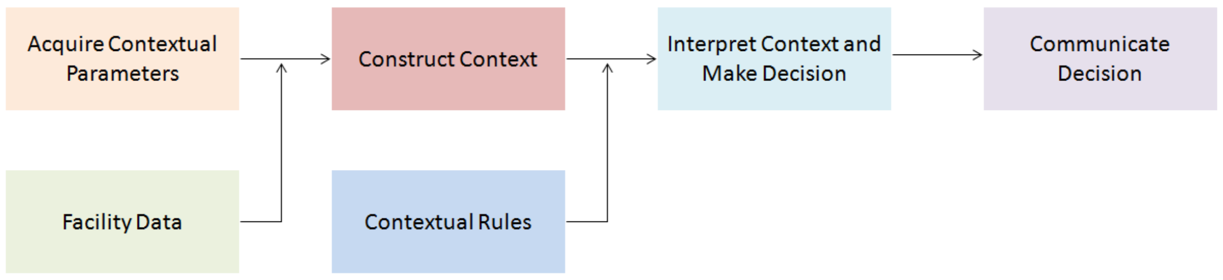


Figure 2.1: Framework for developing context-aware applications in the civil infrastructure domain

2.2.1 Acquiring Contextual Parameters

Civil infrastructure entities that could benefit from support via context-aware computing applications largely belong to two categories: 1) human user and 2) machine agents. The framework utilizes four essential categories of contextual information: identity, time, activity status, and spatial information to define the fully qualified context of end human user and machine agents. Contextual parameters can be acquired through either manual data entry or can be automatically sensed using appropriate sensor technology.

Parameters such as names, IDs, organization affiliations, entity types, and other identifiers are used to uniquely identify agents and entities. It is important that entities be assigned a unique set of identifiers within the application's namespace. Monitoring time context information helps characterize situations and enables applications to leverage the value of historical information by placing the on-going operation task in perspective. Time information is typically monitored by establishing time stamp and time period parameters. However, some applications can be designed by monitoring causality and/or precedence (i.e., the relative order of events) (Dey et al 2001).

Activity status information identifies the inherent characteristics of all computational, operation facility, and end user agents involved in the activity being supported by the context-aware application. For a facility, activity status parameters are of two types: 1) environmental parameters, which are imposed externally; and 2) functional parameters, which are internally generated as a reaction to the environmental parameters. For example, in the case of a highway bridge facility that is being inspected, environmental parameters can be the current temperature, climate conditions, and traffic patterns; the internal parameters can be the stress, strains, and vibrations generated by the bridge facility as a result of the environmental factors. For human users, activity status parameters can refer to physiological factors such as vital signs and tiredness, psychological factors such as mood and enthusiasm, or other factors that describe characteristics of individuals and groups. For machines, activity status parameters can refer to factors such as fuel consumed, exhaust volumes, power generated, and productivity. For computational entities, activity status refers to any attribute that can be queried—such as uptime, process time, CPU load, and the state of files in a file system (Dey et al 2001). It may also be important to monitor the progress of operation tasks and the state of the context-aware computing application itself.

Parameters that identify the spatial context of human and machine agents are used to establish proximate selection methodologies. Proximate selection is a technique where the objects that are most spatially relevant are emphasized or otherwise made easier to choose (Schilit et al 1994). For human agents, spatial contextual parameters such as location, head orientation, and eyeball movement can be reconciled with the human user's body structure to define the fully qualified spatial context. For machine agents, spatial contextual parameters such as location and angular articulation of the parts of interest can be reconciled with the machine's design and kinematics to define spatial context.

2.2.2 Physical and Functional Characteristics of Facilities

Context-aware systems need data models to define and store information about the physical and functional characteristics of operational facilities in a machine-processable form. The requirements and goals for designing data models for context-aware computing applications are as follows (Korpipaa et al 2003).

- Simplicity: To ease the application development, the expressions and relations used by data models should be as simple as possible.
- Flexibility and Extensibility: The data model should support the addition/removal of new physical/functional elements and relations.

- Genericness: The data model should not be limited to certain types of elements and relations but must be able to support different types of elements and relations.
- Expressiveness: The data model should be allowed to describe as much physical/functional characteristics as possible in abstract and arbitrary detail.

The most relevant existing data modeling approaches, which are based on the data structures used for representing and exchanging information in the respective applications, include key value models, markup scheme models, graphical models, relational data models, object oriented models, logic based models, and ontology based models. These models were evaluated based on the aforementioned requirements; ontology-based data models were found to be the most expressive data structures and to best fulfill the requirements, followed by logic-based models, object oriented models, and relational data models (Strang and Linnhoff-Popien 2004).

Ontology-based data models represent a description of the elements, concepts, and relationships. Thus, ontology-based models are a very promising instrument for representing facility information due to their inherently rich and formal expressiveness, and due to the possibilities for applying ontology reasoning methods. Several existing context-aware frameworks use ontology-based data models as underlying context models in their applications across various domains. In the civil infrastructure industry, there are two currently existing ontology-based models that can serve this purpose—CIS/2 and BIM—as shown in Figure 2.2. CIS/2 is a product model and electronic data exchange file format for structural steel project information. The use of CIS/2 data models in the structural steel field is a highly mature practice and is used by almost every large-scale structural steel project. BIM is a process involving the generation and management of physical and functional information of operational buildings from the conceptual stage, through design, construction, operational life, and demolition. The use of BIM in the building construction field is a reasonably mature practice that is gaining rapid acceptance among small-scale, medium-scale, and large-scale projects alike. The ubiquitous nature and growing acceptance of CIS/2 and BIM data models in the civil infrastructure industry make them excellent choices to serve as data structures to support and develop context-aware applications in the civil infrastructure domain.

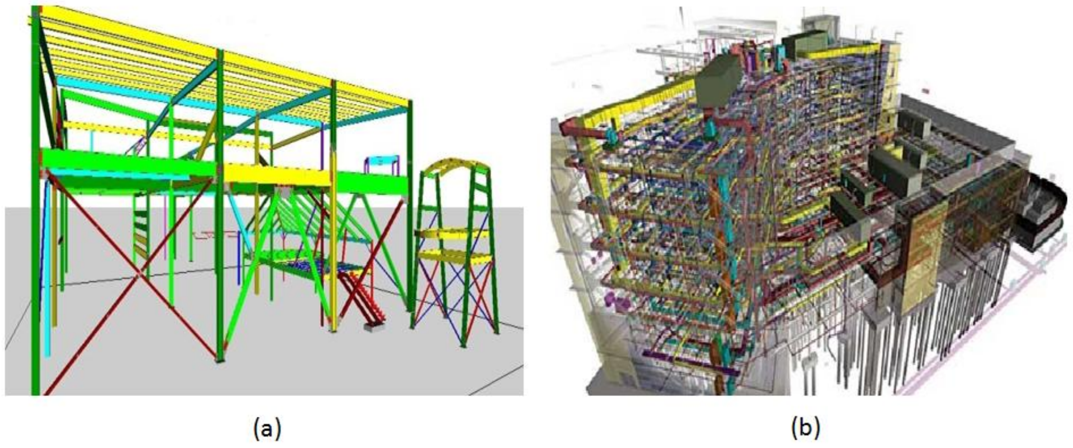


Figure 2.2: (a) CIS/2 (Image courtesy Robert Lipman, NIST) and (b) BIM (Image courtesy Ghafari Associates) ontology-based data models

BIM can also be used as a data model for developing context-aware applications for civil infrastructure facilities such as bridge infrastructure. For example, the controlled drilling system for embeds into bridge decks, described in Chapter 4, utilizes BIM to represent regions of space that are safe/unsafe for drilling. However, the adoption of BIM for such civil infrastructure facilities is difficult in most applications, and researchers are currently pursuing efforts to develop ontology-based data models for infrastructure facilities such as bridges (Bentley 2013; Heikkila and Hovila 2012; Shim et al 2011). Meanwhile, alternative data models can be used to develop context-aware applications for supporting operations in such civil infrastructure facilities. For example, the context-aware bridge inspection system, presented in Chapters 6 and 7, utilizes a relational data model to support bridge inspection operations.

2.2.3 Constructing the Context

There are at least three types of situated data objects that context-aware applications might be interested in to construct their context. The first type of data objects are sensing and computing devices such as video cameras, displays, printers, accelerometers, strain gauges, temperature sensors, databases, and other communicating agents. The second type of objects are located non-physical data objects and services that are typically associated with particular locations. For example, lists of instructions, regulations, and operational guidelines—associated with particular components of infrastructure facilities—can be geo-tagged to appropriate components. The third type of situated data objects are the physical and functional characteristics of facilities that are described in section 2.2.2 (Schilit et al 1994).

Large infrastructure facilities are usually characterized by a large-size data set of situated data objects corresponding to the operations being supported. Proximate selection methodologies are used to prioritize data objects of spatial interest, thus reducing the computational load on the context-aware application. The first step in developing a proximate selection method is to determine the spatial context of the end agent, and this depends on whether the end agent is a human user or a machine, as shown in Figure 2.3. The spatial context of human users is typically the field of view of the human, which is geometrically constructed using the user's location, head orientation, eyeball gaze, and physical characteristics, as shown in Figure 2.3 (a). Methods to construct the field of view of a human user are well documented in computer graphics literature (Assarsson and Moller 1999; Khoury 2009). The spatial context of machine agents is typically the zone of influence of the machine (i.e. the geometry of the machine's components that interact with the operational facility). For example, the zone of influence of an excavator and a drill are the geometry of the excavator bucket's teeth (Talmaki 2012) and the geometry of the drill bit tip, as shown in Figures 2.3 (b) and 2.3 (c), respectively. Based on the sensed spatial contextual parameters and human body (or machine equipment) geometry, the field of view (or zone of influence) of the agent is constructed using principles of forward kinematics (Beggs 1983; Garg and Kamat 2012; Talmaki 2012).

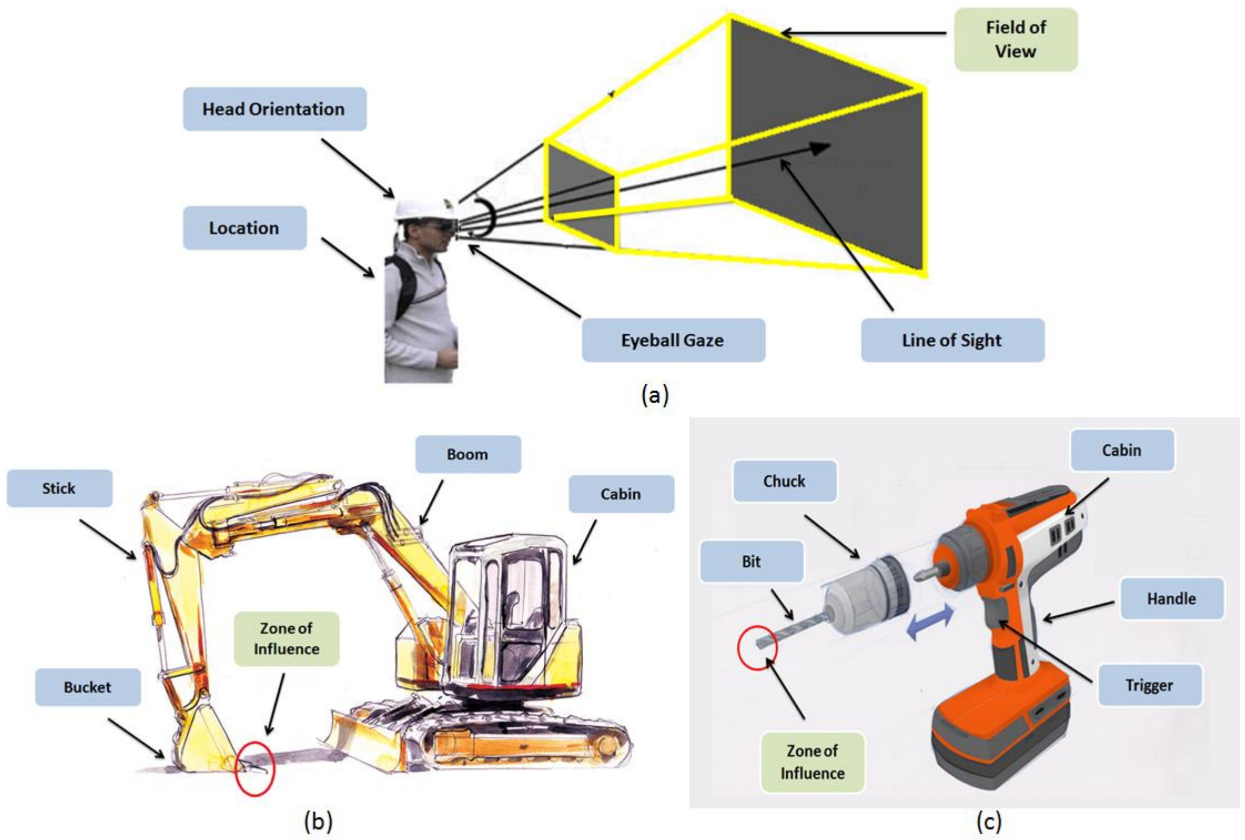


Figure 2.3: The spatial context of (a) human user and machine agents such as (b) excavators and (c) mechanical drills

Proximate selection algorithms involve using a locus as an input to select appropriately situated data objects. The locus is a geometric definition related to the spatial context as defined by the context-aware application developers. For example, the context-aware frameworks used for human user agents typically prioritize the locus as the radial region of space surrounding the line of sight. The locus of the controlled drilling application, described in Chapter 4, might be defined as a cylindrical region of space around the drill bit. The selected data objects are then reconciled with identity, activity status, and time contextual parameters to construct the fully qualified context of the end user agent.

2.2.4 Interpreting Context, Making and Communicating Decisions

Contextual rules are a logically ordered set of if-then guidelines that regulate the behavior of context-aware applications based on the sensed/constructed context. The application algorithm parses the constructed context through the contextual rules to make appropriate decisions as determined by the application developers. The decisions are then communicated to the end human and/or machine agents using suitable techniques in order to achieve desired agent behavior. For example, if the end agent is a human user—such as a bridge inspector, drill operator, or an excavator operator—the decisions are communicated using visual display and audio signal aids. If the end agent is an automatically operated machine, the decisions are communicated to the machine's control system as instructions.

Based on the communicated decisions, the end agents modify their current behavior and adopt the behavior desired by the application. The application must be designed such that the time required for completing the process of acquiring contextual parameters, constructing context, interpreting context, making appropriate decisions, and communicating decisions is subject to hard/soft real-time standards depending on the nature of the operation as described previously in Section 2.1.

2.3 Architecture of Context-Aware Applications

2.3.1 Controlled Drilling for Embeds into Reinforced Concrete Bridge Decks

Chapter 4 describes the development of the drill feedback application and the laser projector-based guidance application to control drilling for embeds into reinforced concrete bridge decks. The architecture of the drill feedback and laser projector-based guidance applications is shown in Figures 2.4 and 2.5, respectively.

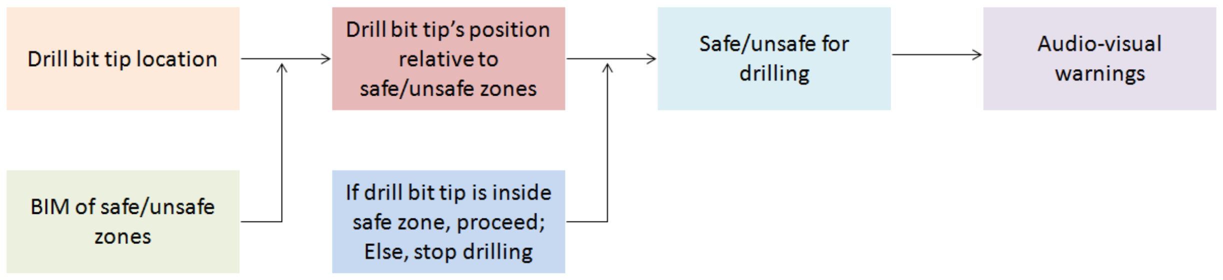

Drill bit tip location
BIM of safe/unsafe zones
Drill bit tip's position relative to safe/unsafe zones
If drill bit tip is inside safe zone, proceed; Else, stop drilling
Safe/unsafe for drilling
Audio-visual warnings

Figure 2.4: Architecture of the drill feedback control application

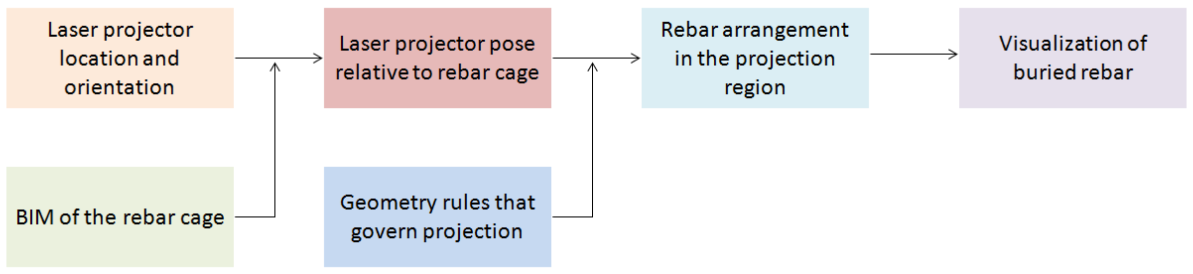

Laser projector location and orientation
BIM of the rebar cage
Laser projector pose relative to rebar cage
Geometry rules that govern projection
Rebar arrangement in the projection region
Visualization of buried rebar

Figure 2.5: Architecture of the laser projector-based guidance application

As described in Sections 4.3.6 and 4.3.7, the contextual parameters tracked by the drill feedback and laser projector-based guidance applications are the pose of the drill bit tip and the laser projector, respectively. The facility data used by the drill feedback and laser projector-based guidance applications are the locations of the safe/unsafe drilling zones and the locations of the rebar, respectively. As described in Sections 4.3.3 and 4.3.4, the facility data is acquired using laser scanning and/or photogrammetry methods, and is stored as BIM. The drill feedback application warns the drilling personnel when they are about to strike rebar or utility lines using audio-visual communication methods. The laser projector-based application guides drilling personnel by helping them visualize the arrangement of the rebar underneath the concrete.

2.3.2 Automated Fault Detection in Operational Buildings

Chapter 5 describes the development of an automated fault detection application for supporting repair operations in case of HVAC system failure. The architecture of the automated fault detection application is shown in Figure 2.6.

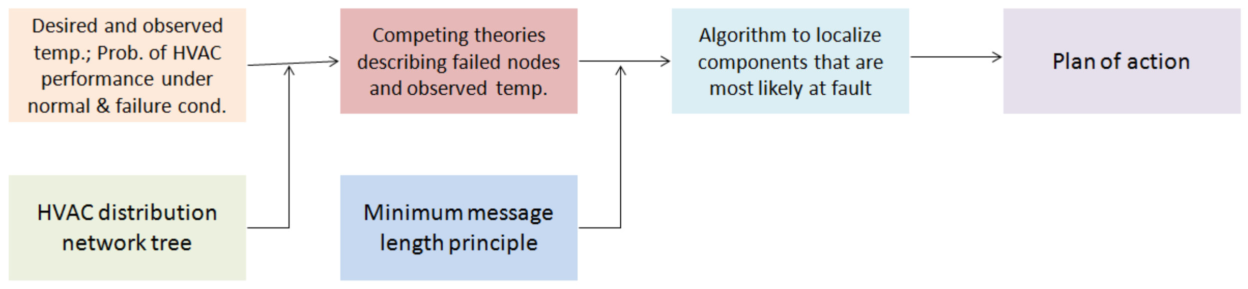

Desired and observed temp.; Prob. of HVAC performance under normal & failure cond.
HVAC distribution network tree
Competing theories describing failed nodes and observed temp.
Minimum message length principle
Algorithm to localize components that are most likely at fault
Plan of action

Figure 2.6: Architecture of the automated fault detection application for HVAC repair

The facility data relevant to the application domain is the HVAC system distribution network. As described in Section 5.4.1, the application extracts the HVAC distribution network model from the BIM corresponding to the facility and stores it as a tree network model. As described in Sections 5.4.2 and 5.4.3, the fault detection application tracks the desired and observed temperatures in the occupant rooms as contextual parameters. Additional contextual parameters used by the application include the probability distribution of the HVAC performance at the occupant rooms for two cases—when the occupant room is affected by HVAC failure, and when it is unaffected. The algorithms used by the application to determine the HVAC components that are most likely at fault and to generate the plan of action are described in Sections 5.4.4 and 5.4.5. The fault detection application communicates the plan of action using the graphical user interface of the BIM plug-in as illustrated in Section 5.4.5.

2.3.3 Context-Aware Bridge Inspection Routines

Chapter 7 describes the development of a context-aware bridge inspection solution for supporting condition assessment decision-making. The architecture of the context-aware bridge inspection solution is shown in Figure 2.7.

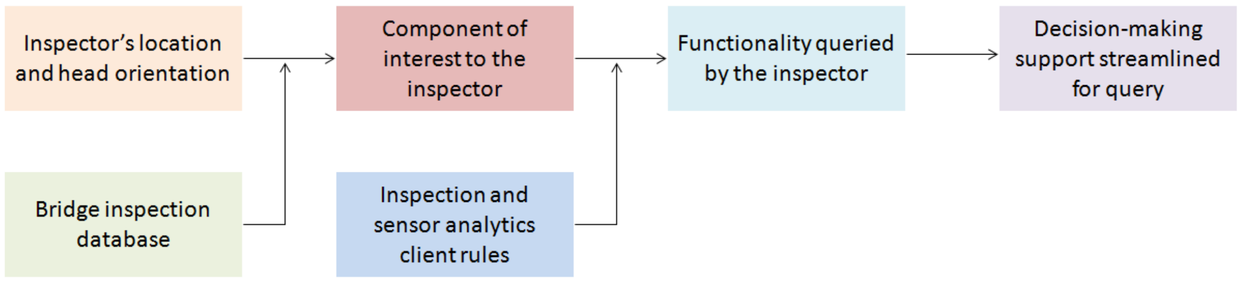

Inspector's location and head orientation
Component of interest to the inspector
Functionality queried by the inspector
Decision-making support streamlined for query
Bridge inspection database
Inspection and sensor analytics client rules

Figure 2.7: Architecture of the context-aware bridge inspection solution

As described in Section 7.2.2, the facility data relevant to the application is stored in a PostgreSQL database and it includes the bridge inventory data, inspection information, and sensor information. The inspector's spatial-context is constructed by continuously tracking the mobile inspector's location and head orientation, as described in Section 7.2.4. The application reacts to the sensed context and allows the inspector to query embedded functionality. Upon querying, the application uses its graphical user interface to display streamlined condition assessment information, as described in Sections 7.3.2 and 7.3.3.

2.4 Summary and Conclusions

This chapter presented a classification of real-time systems based on the strictness of deadlines and the impact of failing to meet them. The chapter then described the overarching framework for developing context-aware applications in the civil infrastructure domain. The framework acquires contextual parameters, accesses facility data and acquired contextual parameters to construct the context, interprets the constructed context based on pre-defined rules, makes appropriate decisions, and communicates them to the end user agent.

The chapter then discusses four essential categories of contextual information—identity, time, activity status, and spatial information—that define the fully qualified context of civil infrastructure agents. The chapter then reviews the requirements and goals for designing data models for context-aware computing applications, and evaluates standard civil infrastructure product models based on the guidelines. Ontology-based data models such as BIM and CIS/2 are found to be well suited for the purpose of developing context-aware applications. The use of relational data models is suggested for facilities that do not have existing ontological data models to comprehensively represent their physical and functional characteristics. The chapter then defines spatial-context for end human and machine agents typically encountered in the civil infrastructure domain. Methods to construct the spatial context of civil infrastructure agents are then presented. The constructed spatial context is reconciled with non-spatial contextual parameters to define the fully qualified context, which is then used to interpret context, make desired decisions, and communicate them to the end user agent.

It is important to note that the framework presented in this chapter is independent of any specific technology and can be used as a generic guideline to introduce context-awareness into real-time decision-making applications in civil infrastructure and related domains. Chapters 3 through 7 demonstrate the power of the developed framework by presenting specific scenarios where the framework is used to design context-aware applications that address the issues and problems pertinent to the scenarios.

2.5 References

Adams, T.M., Juni, E., Siddiqui, M.K., and Dzienkowski, J.E. (2005). "Integrated Field and Office Tools for Bridge Management," Transportation Research Record, Transportation Research Board, Vol. 1933(1), pp. 35–43.

Akula, M., Lipman, R.R., Franaszek, M., Saidi, K.S., Cheok, G.S., and Kamat, V.R. (2013). "Real-Time Monitoring: Augmenting Building Information Modeling with 3D Imaging Data to Control Drilling for Embeds into Reinforced Concrete Bridge Decks," Automation in Construction (In Review).

Assarsson, U., and Möller, T. (1999). “Optimized View Frustum Culling Algorithms,” Technical Report 99-3, Department of Computer Engineering, Chalmers University of Technology, Sweden, pp. 1–29.

Baldauf, M., Dustdar, S., and Rosenberg, F. (2007). “A Survey on Context-Aware Systems,” Int. Journal of Ad Hoc and Ubiquitous Computing, Vol. 2(4), pp. 263–277.

Beggs, J.S. (1983). Kinematics, 1st Edition, Hemisphere Publishing, Washington D.C., ISBN: 0-89116-355-7, pp. 1–168.

Bentley (2013). “About Bridge Information Modeling” – Website of Bentley Systems, Incorporated <http://www.bentley.com/en-US/Solutions/Bridges/brim/> as accessed on 17th April, 2013.

Biegel, G., and Cahill, V. (2004). “A Framework for Developing Mobile, Context-Aware Applications,” Proc. of the 2nd IEEE Annual Conference on Pervasive Computing and Communications (PerCom 2004), Institute of Electrical and Electronics Engineers (IEEE), Orlando, Florida, pp. 361–365.

BridgeInspect™ (2012). “BridgeInspect Software for Bridge Inspectors and Managers” – Website of BridgeInspect™ <http://www.bridgeinspect.com/> as accessed on 15th December, 2012.

BridgeWorks.NET (2012). “Washington State Department of Transportation's Bridge Inspection and Reporting Software” – Website of the Washington State Department of Transportation <http://www.wsdot.wa.gov/LocalPrograms/Bridge/BridgeWorks.htm> as accessed on 15th December, 2012.

Dey, A.K., Abowd, G.D., and Salber, D. (2001). “A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications,” Human–Computer Interaction, Vol. 16(2-4), pp. 97–166.

Efstratiou, C., Cheverst, K., Davies, N., and Friday, A. (2001). “An Architecture for the Effective Support of Adaptive Context-Aware Applications,” Lecture Notes in Computer Science, Springer, pp. 15–26.

Garg, A., and Kamat, V.R. (2012). "Virtual Prototyping for Robotic Fabrication of Rebar Cages in Manufactured Concrete Construction," Proc. of the 2012 Conference on Construction Applications of Virtual Reality (CONVR), National Taiwan University, Taipei, Taiwan, pp. 309–318.

Golabchi, A., Akula, M., and Kamat, V.R. (2013). “Leveraging BIM for Automated Fault Detection in Operational Buildings,” The 30th Int. Symp. on Automation and Robotics in Construction and Mining (ISARC 2013), Montreal, Canada (In Press).

Heikkilä, R., and Hovila, J. (2012). “National Guidelines for Bridge Information Modeling and Automation,” Gerontechnology, International Society for Gerontechnology (ISG), Vol. 11(2), pp. 66–68.

Indiana State BIAS (2012). “Indiana Bridge Inspection Application System” – Website of the Indiana State Bridge Inspection Application System <https://inbridges.indot.in.gov/> as accessed on 15th December, 2012.

Khaitan, S.K., and Gupta, A. (2013). “Priority-Based Resource Allocation,” High Performance Computing in Power and Energy Systems, 2013 Edition, Section 4.4, Springer Berlin Heidelberg, Berlin, Germany, ISBN: 978-3-642-32682-0, pp. 141–142.

Khoury, H.M. (2009). "Context Aware Information Access and Retrieval for Rapid On-Site Decision Making in Construction, Inspection and Maintenance of Constructed Facilities," PhD Dissertation, Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI.

Korpipaa, P., Mantyjarvi, J., Kela, J., Keranen, H. and Malm, E-J. (2003). “Managing Context Information in Mobile Devices,” IEEE Pervasive Computing, Institute of Electrical and Electronics Engineers (IEEE), Vol. 2(3), pp. 42–51.

Schilit, B., Adams, N., and Want, R. (1994). “Context-Aware Computing Applications,” 1st Workshop on Mobile Computing Systems and Applications (WMCSA 1994), Institute of Electrical and Electronics Engineers (IEEE), 85–90.

Shim, C.S., Yun, N.R., and Song, H.H. (2011). “Application of 3D Bridge Information Modeling to Design and Construction of Bridges,” Procedia Engineering, Elsevier Science, New York, NY, Vol. 14, pp. 95–99.

Strang, T., and Linnhoff-Popien, C. (2004). “A Context Modeling Survey,” 1st Int. Workshop on Advanced Context Modeling, Reasoning, and Management, 2004 ACM Int. Joint Conference on Pervasive and Ubiquitous Computing (UbiComp 2004), Association for Computing Machinery (ACM).

Talmaki, S. (2012). "Real-Time Visualization for Prevention of Excavation Related Utility Strikes," Ph.D. Dissertation, Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI.

Chapter 3

Integration of Infrastructure-Based Localization Systems with Inertial Navigation for Ubiquitous Localization

3.1 Introduction

Context-aware computing is defined as the use of environmental characteristics such as a user's location, time, identity, profile and activity to inform the computing device so that it may provide information to the user that is relevant to the current context (Burrell and Gay 2001). Context-aware computing can potentially enable mobile users in a wide variety of fields to leverage knowledge about various context parameters to ensure that they get highly specific information, pertinent to the decisions at hand. The relevance of context awareness for mobile users has been demonstrated in several engineering applications (Aziz et al 2005). The concept of context-aware information delivery centers around the creation of a user-centered, mobile, dynamic (indoor and outdoor) computational work environment that has the ability to deliver relevant information to on-site mobile users by intelligent interpretation of their characteristics in space and time so that they can take more informed decisions. Context awareness can thus be of great value for civil engineering inspectors, emergency responders, security and military personnel. For example, interpreting the context of civil engineers during post disaster reconnaissance, or while conducting a bridge inspection, can allow bi-directional flow of streamlined information and thereby improve the efficiency of the decision-making process.

Context-aware applications can be used in providing support to complex, tedious, and time-consuming tasks. Civil engineers, fire fighters, military personnel and a host of other professionals stand to benefit from context-aware applications as they make the bi-directional flow of information more efficient and relevant based on a mobile user's context. Bridge inspections, for example, are currently documented manually—the bridge inspector assesses the condition of a bridge based on standard rating guidelines and previous bridge inspection reports (Farrar 2011). The inspector carries the rating guidelines and the previous reports in their paper form along with the current report forms while conducting the inspection. In many cases, the inspector has to come to the site with excessive preparation, and plenty of time and effort is wasted in searching, streamlining, and retrieving relevant information. Upon returning to the office, after the inspector completes the inspection, the relevant data gathered on the field is entered into a database management system. However, context-aware computing can tremendously reduce the time and effort involved in conducting such bridge inspections by facilitating the bi-directional flow of information between the database management system and the on-site inspector. Based on the context of the inspector, streamlined data (such as the relevant parts of rating guidelines and previous inspection reports) can be supplemented to field inspectors to support their operations. The process of updating the database management system with appropriate field data can also be similarly automated.

3.1.1 Importance of the Research

Prior applications of context-aware computing have included fieldwork (Kortuem et al 1999), museums (Fleck et al 2002), route planning, libraries (Aittola et al 2003), and tourism

(Laukkanen et al 2002). Examples of other projects that have specifically focused on location-based information delivery have included the GUIDE project (Davies et al 1999) and the Mobile Shadow Project (Fischmeister et al 2002). Previous work on context-aware computing also involved the development of mechanisms for information support on construction sites (Cox et al 2002; Singhvi et al 2003; Bowden et al 2004; May et al 2005; Aziz et al 2005; Aziz et al 2006; Skattor 2007).

The ability to automatically access and retrieve accurate information that is of decision-making context at an arbitrary time and location can significantly increase the productivity of constructors, engineers and inspectors. Spatial context is one of the most important parameters considered in context-aware computing. In our research group's prior work, a framework for the high-precision identification of contextual information in location-aware engineering applications has been developed based on a mobile user's location and head orientation (Khoury and Kamat 2009). For such context-aware engineering applications, the ability to ubiquitously track a mobile user's location continuously and accurately is of utmost importance. Several infrastructure-based localization systems have been developed and are commercially available for outdoor localization (Global Positioning System (GPS)) and indoor localization (Wireless Local Area Networks (WLAN), Ultra-Wide Band (UWB) and Indoor GPS (iGPS)).

The main drawback of the aforementioned location tracking technologies is their dependency on pre-installed infrastructure, which makes them unsustainable in a dynamic environment that cannot be prepared or retrofitted, apriori. To overcome this drawback, several infrastructure-independent localization systems have been developed in recent years. Typically, infrastructure-independent localization systems are based on inertial measurements and make use of high-accuracy gyroscopes and accelerometers. The main drawback of infrastructure-independent localization systems based on inertial measurements is the accumulated drift error that grows with the distance travelled by the mobile user (Ojeda and Borenstein 2007).

The objective of this chapter is to present research that developed the architecture of an Integrated Tracking System (ITS) that seamlessly integrates several infrastructure-based localization systems with infrastructure-independent localization technologies to continuously and effectively track the current location and trajectory of a mobile user in dynamic environments. The ITS effectively uses infrastructure-based localization systems when in range and bridges the gap between such localization systems—when out of range—through infrastructure-independent localization technologies. Simultaneously, the ITS uses infrastructure-based localization systems—which are typically more accurate—to correct the drift errors accumulated in infrastructure-independent localization technologies. An implementation of the ITS, based on integrating an infrastructure-based localization system (GPS) with an infrastructure independent localization system—the Personal Dead Reckoning (PDR) system (Ojeda and Borenstein, 2007) —is described. The aforementioned tracking system is particularly useful in post-disaster scenarios where pre-installed infrastructure may be partially or completely damaged, or where preparing and retrofitting the environment a priori is not feasible. In context-aware applications that support computing tasks in non-emergency scenarios, the user's location is typically tracked using an infrastructure based localization system. Infrastructure-based localization systems are typically expensive to install, maintain and replace. The chapter then presents an ITS based on integrating the PDR system with architectural markers—installed at known locations—in the environment. The author presents the developed techniques to correct the drift errors accumulated by the PDR system based on a priori knowledge of the location of specific marker points in the indoor environment. Finally, the chapter presents a hybrid localization method that improvises on the PDR system by using the mobile user's intelligence and awareness of the environment. The aforementioned tracking system is particularly useful where installing 'marker points' in the environment is not feasible. The tracking system uses natural architectural features of the environment as proxy marker points, and uses an embedded visualization application to correct the drift errors accumulated by the PDR system.

3.2 Current State of Knowledge

3.2.1 Infrastructure-Based Tracking Systems

Real-Time Kinematic – Global Positioning System (RTK-GPS) is a convenient option to track a mobile user continuously in an outdoor environment. It is highly accurate and is free of accumulated errors (Cobb 1997). RTK satellite navigation is a technique used in land survey and in hydrographic survey based on the use of carrier phase measurements of the GPS where a single reference station provides the real-time corrections, providing up to centimeter-level accuracy. The typical nominal accuracy for these RTK-GPS systems is 2.5 cm horizontally and 5 cm vertically.

Integrated surveying techniques employ RTK-GPS to provide surveyors with greater flexibility and control over how to perform surveys. Integrated surveying techniques that complement RTK-GPS with conventional surveying methods have resulted in a significant increase in surveying productivity (Lemmon and Wetherbee 2005). Automated steering technology based on

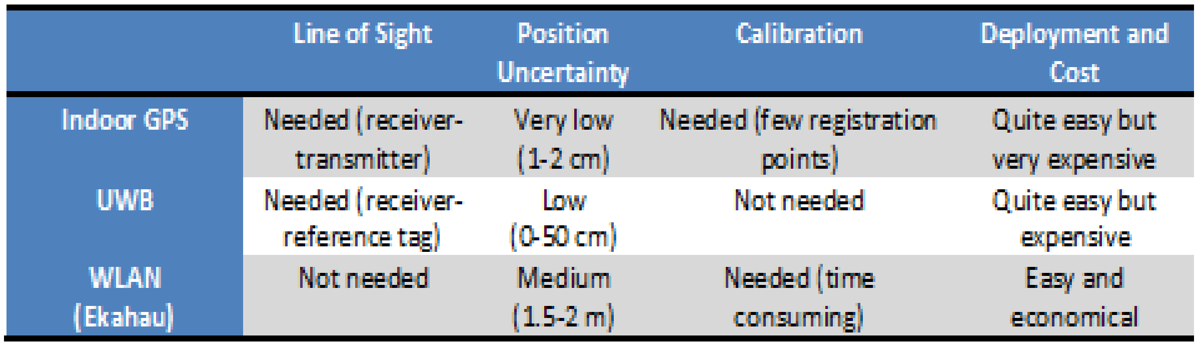
RTK-GPS is used by crop growers to control heavy, fast-moving agricultural machines (Buick 2006). RTK-GPS is proposed to be used to facilitate navigation for precise Intelligent Transportation System services, for instance, precise navigation, autonomous driving, lane-based traffic or fleet management, lane-based road use charging, and law enforcement (Meng et al 2008).

RTK-GPS is used for construction plant control and guidance. RTK-GPS systems have been deployed to allow real-time centimeter localization that allows bulldozer drivers to operate the machinery in a semi-autonomous manner. The SiteVision GPS System is able to provide real-time guidance for a bulldozer and produce a finished surface that on average is within 3 cm of the design. The accuracy of the system is, however, affected indirectly by external factors such as soil type and weather conditions (Roberts et al 2002). Laser levels are commonly used in civil engineering to replace traditional leveling procedures and to automate construction operations. RTK-GPS provides an alternative to laser levels for both leveling procedures and in automation and has the potential to provide precision on the order of a few millimeters (Roberts et al 2000). However, RTK-GPS being a satellite-based navigation system works very well outdoors but lacks support indoors and is unreliable in dense foliage, in so-called "urban canyons," and generally in any environment where a clear line of sight to the satellites is unavailable.

In recent years, the need for indoor localization has been rapidly expanding in many fields and currently offers significant potential on construction sites in particular. However, unlike outdoor areas, the indoor environment imposes different challenges on location discovery due to the dense multipath effect and building material dependent propagation effect (Khoury and Kamat

2008). There are many potential technologies and techniques that have been suggested to offer the same functionality as a GPS indoors, such as WLAN, UWB, and iGPS.

Typically, in such localization systems, users are tagged with appropriate receivers/tags and a number of nodes (access points, receivers, transmitters, etc.) are deployed at fixed positions indoors. The user then ‘fingerprints’ the indoor environment by recording and storing (in a database) the signal strength corresponding to several installed transmitters as received by the user at several known locations (Ekahau 2007). Once the indoor environment is calibrated, the location of a mobile tagged user can conceptually be determined and tracked continuously by recording the signal strength corresponding to several pre-installed transmitters as received by the mobile user. The received signal strength is matched with the pre-recorded signal strengths at several known locations stored in the database, and the location is determined by implementing triangulation and interpolation algorithms. A detailed comparison of the WLAN, UWB, and iGPS systems has also been done in a recent study (Khoury and Kamat 2008) and is summarized in Figure 3.1.



	Line of Sight	Position Uncertainty	Calibration	Deployment and Cost
Indoor GPS	Needed (receiver-transmitter)	Very low (1-2 cm)	Needed (few registration points)	Quite easy but very expensive
UWB	Needed (receiver-reference tag)	Low (0-50 cm)	Not needed	Quite easy but expensive
WLAN (Ekahau)	Not needed	Medium (1.5-2 m)	Needed (time consuming)	Easy and economical

Figure 3.1: Comparative summary of indoor localization technologies (Khoury and Kamat 2008)

Radio frequency identification (RFID)-based tracking systems (Pradhan et al 2009) and ultrasonic location systems (Hazas and Hopper 2006) have also been developed and implemented under real operating conditions. A comparative study of several indoor and outdoor localization systems—including assisted Global Navigation Satellite Systems (GNSS), Bluetooth-based localization systems, iGPS, laser-based localization systems, pseudolites using GNSS-like signals, ultrasound-based localization system, and WLAN localization systems— that share the market has also been conducted (Mautz, 2009). The study details the accuracy, operation range, and drawbacks (such as low accuracy, sophisticated infrastructures, limited coverage area, and inadequate acquisition costs) of the surveyed systems (Mautz, 2009).

As mentioned previously, the main drawback of the aforementioned tracking technologies is their dependency on pre-installed infrastructure and, in some cases, pre-calibration for fingerprinting. In addition, most technologies are specific to environment (outdoors and indoors). Such dependency makes them unreliable in dynamic environments like construction sites due to constant changes in the site layout. Furthermore, every potential environment cannot be expected to have pre-installed infrastructure and pre-calibration done for fingerprinting. In applications such as post-disaster reconnaissance, any pre-installed infrastructure may itself be partially or completely damaged. It is therefore critical to have a comprehensive location tracking system that can be used reliably irrespective of the mobile user’s environment, and that does not rely entirely on tracking technologies that are dependent on pre-installed infrastructure and pre-calibration techniques.

3.2.2 Infrastructure Independent Tracking Systems

In GPS-denied environments (such as tunnels, urban canyons, etc), ground vehicle navigation systems make effective use of odometry, heading, vehicle dynamic models, and map matching (Dissanayake et al 2001; Forssell et al 2002) for localization. Infrastructure-independent indoor localization systems for mobile human users have been developed using foot-mounted accelerometers with high-accuracy heading sensors (Collin et al 2003), and motion sensors mounted on helmets (Beauregard and Haas 2006). A light-weight infrastructure-independent localization system (PDR system) —based on high-accuracy inertial measurement units and sophisticated step detection algorithms—that is largely independent of the gait or speed of the mobile user has also been developed (Ojeda and Borenstein 2007).

The PDR system is based on inertial navigation and is independent of pre-installed infrastructure and calibration. Although less accurate than WLAN, UWB, and Indoor GPS, it provides sufficient accuracy that degrades gracefully with extreme modes of legged locomotion (Ojeda and Borenstein 2007). The PDR system uses data from the accelerometers and gyroscopes in the Inertial Measurement Unit (IMU) sensor attached to the mobile user's boots. From this data the system computes the complete trajectory of the boot during each step.

The PDR system uses a high-quality, small-sized, light, nano IMU strapped to the side of the mobile user's foot, as shown in Figure 3.2. The IMU is connected to an embedded computer through an RS-422 communication port. The IMU is powered using a small external 7.8-Volt Lithium Polymer battery, making the whole system portable. The computer runs the Linux operating system patched with a real-time extension (Ojeda and Borenstein 2007).

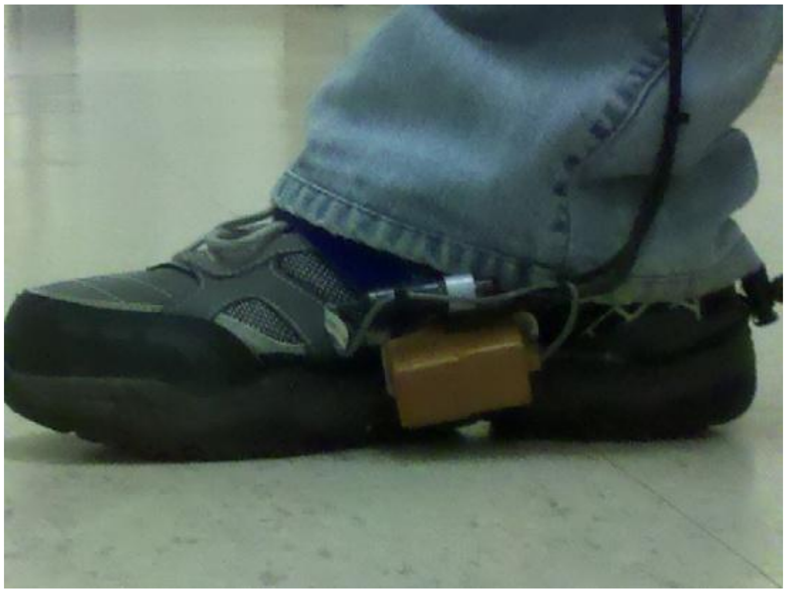


Figure 3.2: The small-sized nano IMU strapped onto a mobile user’s foot

The IMU-based PDR system is very accurate in measuring linear displacements (i.e., distance travelled, a measure similar to that provided by the odometer of a car) with errors being consistently less than 2% of the distance travelled (Ojeda and Borenstein 2007). The accuracy of the PDR system, however, degrades gracefully with extreme modes of legged locomotion, such as running, jumping, and climbing. The main drawback of the PDR system is the drift error that accumulates with the distance travelled by the mobile user.

Another sensor-based pedestrian tracking system that is independent of any infrastructure has been designed where information about human locomotion is monitored by a sensor module composed of accelerometers, gyroscopes, and magnetometers (Huang et al 2010). The acquired sensor readings are used by an algorithm to accurately compute the location of a pedestrian. Through the application of human kinetics, the algorithm integrates two traditional technologies—strap-down inertial navigation and pedestrian dead reckoning. The drift error on the horizontal plane was found to be within 2% of distance traveled (Huang et al 2010).

A wearable pedestrian indoor localization system with dynamic location correction that combines dead reckoning and fiduciary marker-based localization schemes, exclusively using widely available, inexpensive, low-power consumer hardware components has also been developed (Torres-Solis and Chau 2010). The system has an indoor localization accuracy around 3.38% of the total distance walked. This accuracy is comparable to those obtained with solutions deploying specialized high-cost hardware components (Torres-Solis and Chau 2010). Inertial sensing and sensor network technology have also been combined to create a pedestrian dead reckoning system. The dead reckoning performance is further enhanced by wireless telemetry and map matching algorithms (Fang et al 2005).

The inertial navigation system used in this research—to track the mobile user's location—is the PDR system (Ojeda and Borenstein 2007) mentioned above. The PDR system computes the location of the mobile user in a three step process: 1) location estimation, 2) step detection, and 3) Zero Velocity Update (ZUPT). The PDR computes the location as the integral of velocity over time. The velocity is estimated by integrating the accelerations (recorded by the IMU sensors) in the navigation frame. The PDR detects footfalls (the duration of time in which the instrumented foot is in contact with the ground) by analyzing the angular velocity vectors measured by the gyroscopes in the IMU. The PDR then implements the ZUPT technique to limit the drift accumulated during every step (time between two consecutive footfalls). ZUPT is a method of counteracting drift and is commonly used in underwater navigation. ZUPT assumes that the velocity of the mobile user's foot approaches zero, at least once, during a footfall (i.e., no slippage condition). Thus, the PDR expects the velocity vector to show zero readings during a footfall. If the reading is not zero, the PDR assumes that the difference between zero and the momentary reading is the result of accumulated errors during the step interval, and corrects it.

Any of the aforementioned inertial tracking systems (Beauregard and Haas 2006; Collin et al 2003; Fang et al 2005; Huang et al 2010; Torres-Solis and Chau 2010) can be used as an alternative to the PDR. These systems typically use a combination of accelerometers and magnetometers for inertial navigation localization. Step length is typically estimated from accelerometer readings, and advanced Kalman Filter techniques are used to reduce the effect of magnetic disturbances. The research, purposely, does not aim at improving the step length estimation algorithms nor does it aim at improving Kalman Filter techniques. Instead, the ITS enhances the localization ability of the inertial navigation system by integrating it with infrastructure-based localization technologies and adding domain knowledge.

One of the primary ideas this research exploits is that prior knowledge of the true position of certain points along the path of the mobile user can be used to eliminate the drift error accumulated by the PDR system during the mobile user's walk in reaching those points. The location of such points can be obtained from highly accurate infrastructure-based localization systems. The location as determined by infrastructure-based localization is not the true location of the point and contains errors inherent to the localization system being used. However, it can for all practical purposes be considered the true location, especially when the drift accumulated by the PDR system is relatively large compared to the inherent error in the infrastructure-based localization system being used. Thus, by complementing the PDR system with high-accuracy infrastructure-based localization systems, the drift error accumulated in the PDR system can be eliminated. The following section in this chapter describes the architecture of the ITS that seamlessly integrates the PDR system with several infrastructure-based localization systems, and the developed algorithms that correct the drifting error accumulated over time.

3.3 Technical Approach for Ubiquitous Integrated Localization

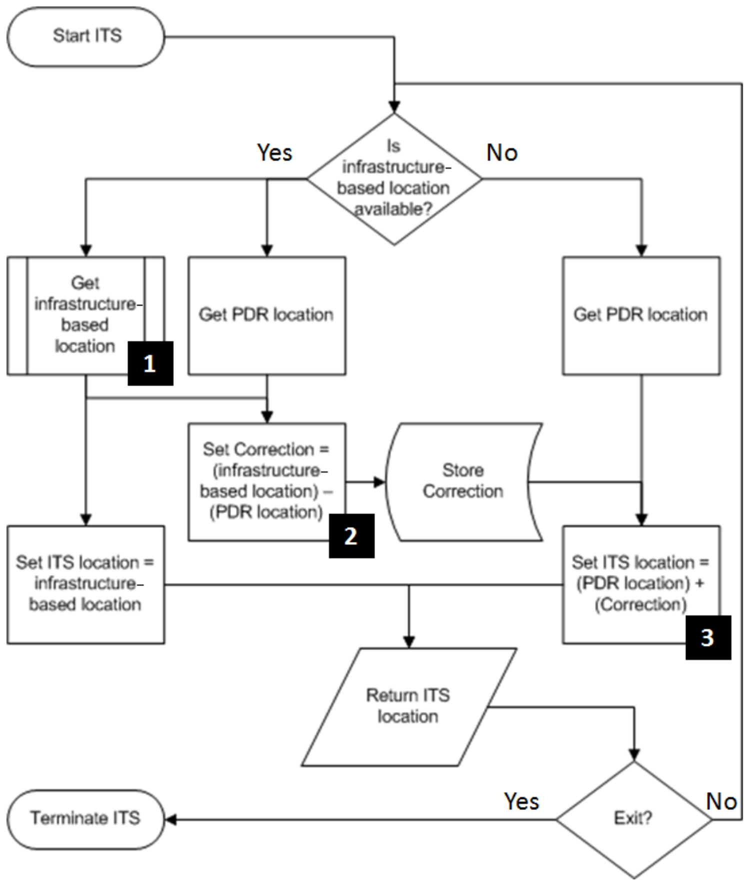
In this research the authors have developed an algorithm that seamlessly integrates several infrastructure-based location tracking systems with infrastructure independent location tracking systems. Localization systems based on technologies like RTK-GPS, UWB, ultrasound systems, iGPS, etc. have relatively low uncertainty in the reported location and are independent of the distance travelled by the mobile user. During the initial stages of tracked navigation, the accumulated drift in infrastructure-independent localization systems like PDR might be lower than the uncertainty in location as determined by the aforementioned infrastructure-based localization systems. However, after travelling enough distance the drift accumulated by infrastructure-independent localization systems starts to overshoot the uncertainty present in the infrastructure-based localization systems.

The ITS uses the concept that when the mobile user is within the range of high-accuracy infrastructure-based location technologies, the user’s location is determined by the most accurate among the available tracking technologies; and when the mobile user moves outside the range of all such infrastructure-based localization systems, the location is determined by using infrastructure-independent location tracking systems. During the time periods when the user’s location is being tracked by infrastructure-independent systems, user intervention and visual domain knowledge are utilized to further reduce the accumulated drift, which would otherwise be corrected only after the next availability of an infrastructure-based localization option.

3.3.1 Integrated Tracking System Algorithm

The overarching algorithm of the ITS is illustrated as a flowchart in Figure 3.3. The mobile user initializes the ITS at the beginning of a tracked navigation. When the user is within the range of infrastructure-based localization systems, the ITS retrieves the mobile user's best known location using the most accurate available infrastructure-based localization option. When available, the location retrieved by the most accurate infrastructure-based localization system in range is considered by the ITS to be the 'true' location of the mobile user, and the ITS returns this position as the user's location as shown in Box 1 in Figure 3.3.

However, the error in this tracked location would equal the inherent uncertainty of the infrastructure-based localization system invoked by the ITS. The process of retrieving the best-known user location using multiple infrastructure-based localization systems is illustrated in Figure 3.4. The mobile user navigates carrying several receivers corresponding to the infrastructure-based localization systems. These receivers are connected to a mobile-computer that serves as a computation center. The first step in retrieving the best known non-PDR localization of the mobile user, as shown in Figure 3.4, is to identify all infrastructure-based localization systems present in the mobile user's range. The next step is to prioritize the identified localization systems based on their accuracy. Once the localization systems are prioritized, the localization client of the system with the highest priority is invoked and the best-known location of the mobile user as determined by the particular localization technology is retrieved.



Start ITS
Is infrastructure-based location available?
Yes
No
Get infrastructure-based location
1
Get PDR location
Get PDR location
Set Correction = (infrastructure-based location) – (PDR location)
2
Store Correction
Set ITS location = infrastructure-based location
Set ITS location = (PDR location) + (Correction)
3
Return ITS location
Exit?
Yes
No
Terminate ITS

Figure 3.3: Flowchart of the ITS algorithm

As illustrated in the flowchart in Figure 3.3, the ITS also retrieves the mobile user's location as determined by the PDR system. Initially, the drift accumulated by the PDR might be lesser than the inherent uncertainty in the infrastructure-based localization systems available but this drift soon exceeds the inherent uncertainty as the mobile user continues navigating. The ITS evaluates the difference between the user's best-known location as determined by infrastructure-based localization systems and infrastructure-independent localization (PDR) systems and stores it as a ‘correction’, as shown in Box 2 in Figure 3.3. This correction is continuously updated as long as the user is within the range of an infrastructure-based localization system.

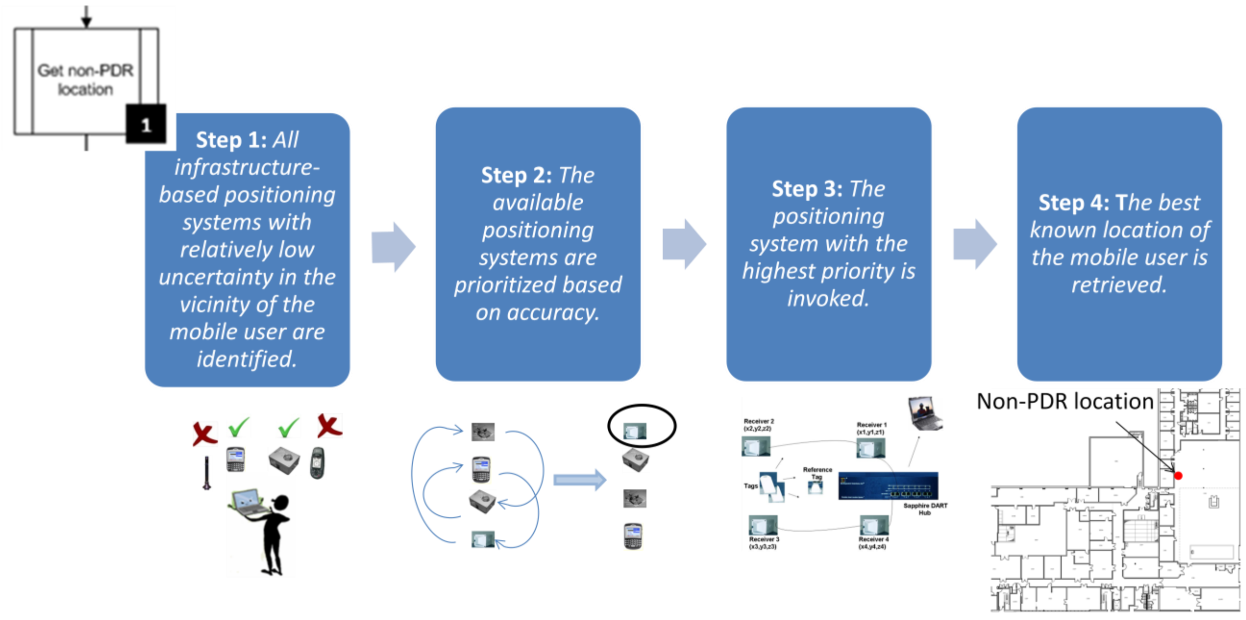

Get non-PDR location
1
Step 1: All infrastructure-based positioning systems with relatively low uncertainty in the vicinity of the mobile user are identified.
Step 2: The available positioning systems are prioritized based on accuracy.
Step 3: The positioning system with the highest priority is invoked.
Step 4: The best known location of the mobile user is retrieved.
Non-PDR location

Figure 3.4: The process of retrieving the best known user location from infrastructure based localization systems

When the user moves out of the range of the infrastructure-based localization systems, the correction is no longer updated continuously, and the correction used by the ITS is the correction in the drift at the last point in the user’s path when infrastructure-based localization systems were still in range. The ITS now determines the mobile user’s location by adding this last known correction to the user’s location as determined by the infrastructure-independent localization (PDR) system, as shown by Box 3 in Figure 3.3. This correction would ensure that prior drift accumulated by infrastructure-independent localization (PDR) —until the very last time the mobile user was within the range of an infrastructure-based localization system—is eliminated and that only the drift accumulated by the infrastructure-independent localization (PDR) system during the subsequent portion of the walk is reflected in the ITS. When the mobile user again moves into the range of a infrastructure-based localization system, the correction is updated once again and the drift error is no longer reflected in the ITS.

For example, consider the situation shown in Figure 3.5 where the mobile user starts navigating in an outdoor environment where RTK-GPS is available and then moves into a GPS-denied environment by entering a building. As long as the mobile user is present in an environment where RTK-GPS is available, the ITS location is determined by RTK-GPS and is within 2.5 to 5 cm (inherent uncertainty in a typical RTK-GPS) of the mobile user's actual path. The ITS continuously updates the drift accumulated by the PDR system in this portion of the walk.

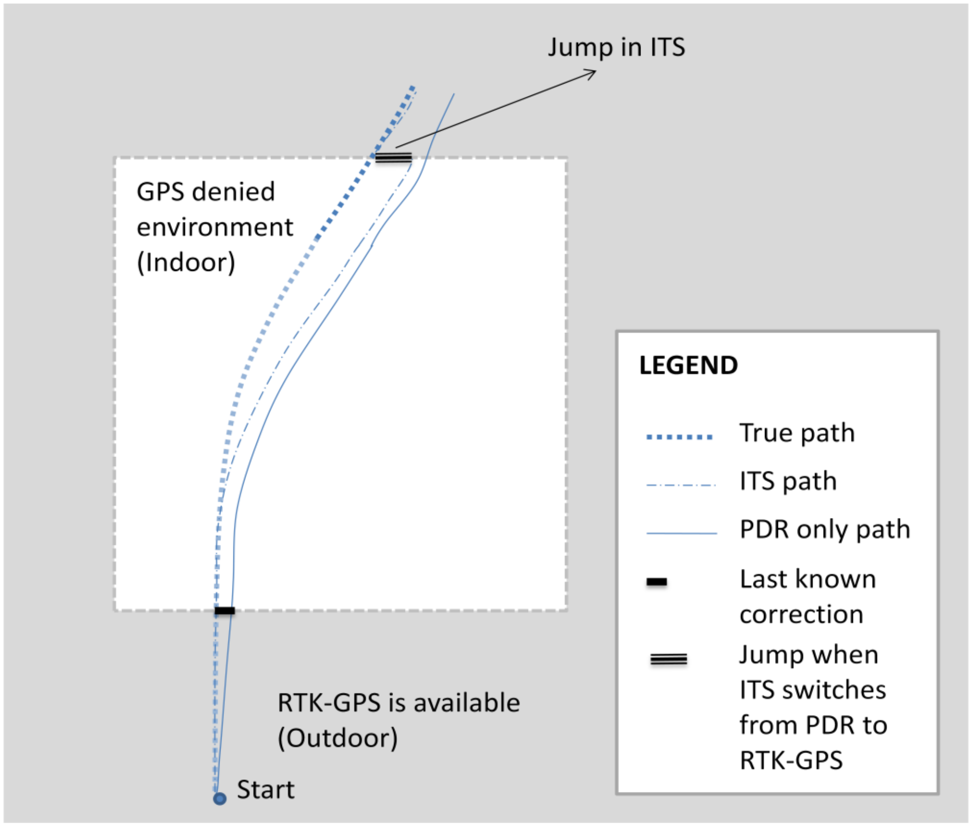

Jump in ITS
GPS denied environment (Indoor)
RTK-GPS is available (Outdoor)
Start
LEGEND
True path
ITS path
PDR only path
Last known correction
Jump when ITS switches from PDR to RTK-GPS

Figure 3.5: Mobile user's path as determined by ITS and PDR compared to the actual path

When the mobile user enters a GPS-denied environment, the user's location is determined by the PDR system. The correction for the drift being accumulated in the PDR system is no longer updated continuously. However, the ITS stores the last known drift correction and applies it to the PDR location at all subsequent points (i.e., throughout the user's path in the GPS-denied environment). Thus, the ITS only reflects the drift accumulated by the PDR system in the GPS-denied environment and not the drift accumulated by the PDR system since the beginning of the walk. When the user re-enters an environment where RTK-GPS is available, the ITS switches its location as determined by the RTK-GPS instead of the PDR system; this is reflected as a "jump" in the user's position coordinates, as shown in the upper portion of Figure 3.5.

3.4 Implementation of the Integrated Tracking System

The ITS has been implemented in three levels, as shown in Figure 3.6. Section 3.4.1 describes the implementation and validation experiments for a version of the ITS that utilizes RTK-GPS localization technology to complement the PDR system. Section 3.4.2 describes the implementation and validation experiments for a version of the ITS that utilizes specific pre-determined indoor correction points to complement and correct the drift accumulated by the PDR system during a mobile user's walk in an indoor (GPS denied) environment. Finally, Section 3.4.3 describes the implementation and validation experiments of a hybrid tracking system that utilizes human intelligence and the ability of the mobile user to discern the environment to complement and correct the drift accumulated by the PDR system during a mobile user's walk in an indoor (GPS denied) environment. The ITS implementation algorithm and code are described in Appendix A.

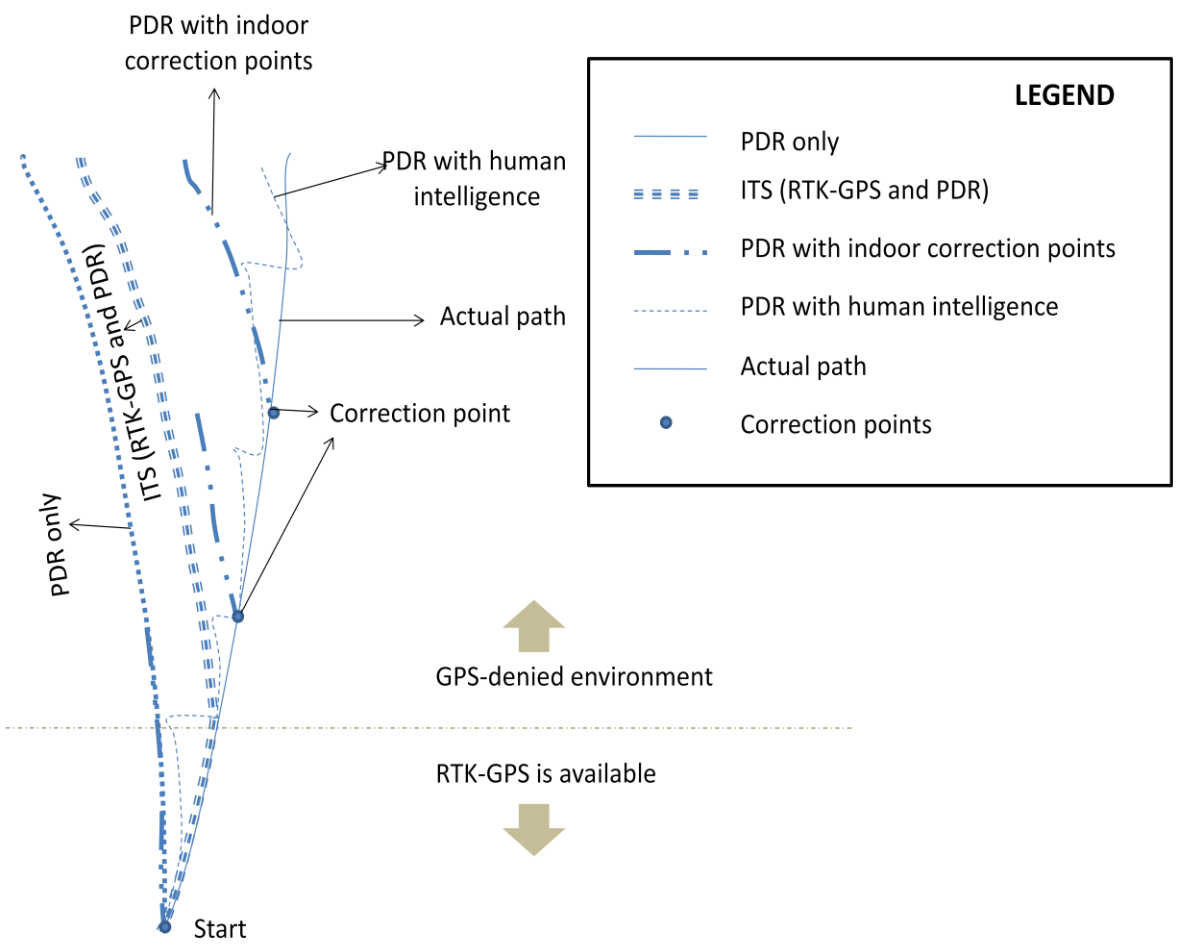


Figure 3.6: Illustration of the drift as reflected by the three levels of ITS implemented

3.4.1 Integrated Tracking System Based on RTK-GPS and PDR

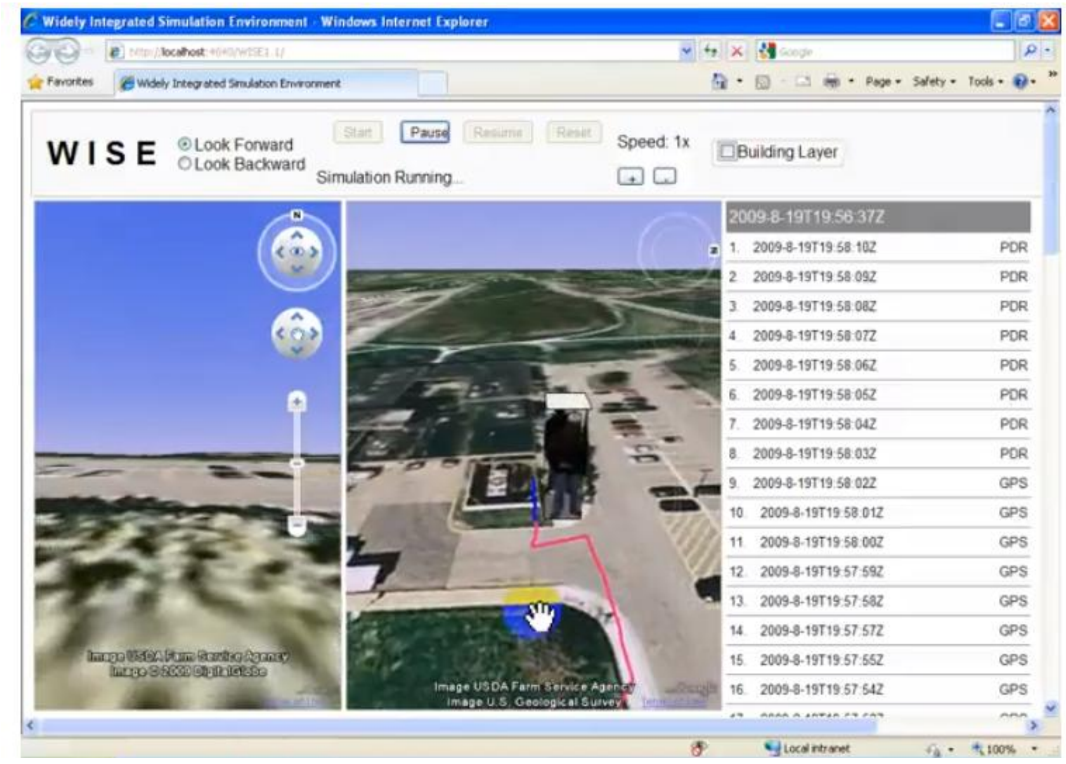
This version of the ITS combines RTK-GPS and PDR systems and minimizes the shortcomings of both technologies by complementing them with each other through integration. The ITS continuously tracks a mobile user and retrieves the user’s location to the best possible degree of accuracy. A standardized format of geographic co-ordinates (latitude, longitude, altitude) along with time stamp markings are employed to denote the user’s location. The ITS performs two major functions:

- Provides a PDR system service when RTK-GPS is blocked.

- Corrects the drift error accumulated in the PDR whenever RTK-GPS is available.

The accuracy of RTK-GPS (2.5 cm to 5 cm) is generally higher than the accuracy of the PDR. The principle behind determining the ITS co-ordinates, as mentioned in the preceding sections, is that RTK-GPS co-ordinates, if available, always take precedence over the PDR co-ordinates. The PDR drift is corrected by applying a correction equal to the difference in the mobile user's GPS and the PDR co-ordinates. When the user is disconnected from the GPS (i.e., when the user no longer has clear line of sight to the satellites), the PDR correction is the same as the PDR correction at the last instant the GPS was available. This correction is applied to the PDR until the GPS signal is regained. When the GPS is unavailable, the ITS co-ordinates are determined by the corrected PDR co-ordinates.

Once the GPS signal is regained, the PDR correction is updated and the RTK-GPS co-ordinates determine the ITS co-ordinates. This updated correction manifests as a "jump" in the ITS co-ordinates. This particular version of the ITS uses the Widely Integrated Simulation Environment (WISE) —a JavaScript enabled web application—to help in visualization (Akula et al 2011). WISE is based on the Google Earth API and ASP.NET 2.0. The hybrid trajectory of the mobile user tracked by the ITS is recorded using the Keyhole Markup Language (KML) and stored on the web server. The user can query the location tracking state either online or offline through a web browser enabled with the Google Earth plug-in. On request, the web server retrieves the relevant location, orientation, and timestamp, and posts it back to the browser side. The received data package is further parsed and rendered in the Google Earth virtual environment, as seen in Figure 3.7 (Akula et al 2011). By doing so, the user can visually analyze and confirm the current tracking status and also perform further numerical analysis on the switching between RTK GPS and PDR systems.



2009-8-19T19:56:37Z		
1.	2009-8-19T19:58:10Z	PDR
2.	2009-8-19T19:58:09Z	PDR
3.	2009-8-19T19:58:08Z	PDR
4.	2009-8-19T19:58:07Z	PDR
5.	2009-8-19T19:58:06Z	PDR
6.	2009-8-19T19:58:05Z	PDR
7.	2009-8-19T19:58:04Z	PDR
8.	2009-8-19T19:58:03Z	PDR
9.	2009-8-19T19:58:02Z	GPS
10.	2009-8-19T19:58:01Z	GPS
11.	2009-8-19T19:58:00Z	GPS
12.	2009-8-19T19:57:59Z	GPS
13.	2009-8-19T19:57:58Z	GPS
14.	2009-8-19T19:57:57Z	GPS
15.	2009-8-19T19:57:55Z	GPS
16.	2009-8-19T19:57:54Z	GPS

Figure 3.7: Interface of the Widely Integrated Simulation Environment (WISE)

The implementation and validation experiments pertaining to this version of the ITS focus on three different types of experiments: (1) short and simple walks, (2) short and complex walks and (3) longer walks.

Short and simple walks: Relatively simple walks with duration between 3 and 5 minutes (indoors) are classified as short walks. These walks involve few turns and almost no abrupt disturbances in motion. Table 3.1 summarizes the “jumps” in the user’s position (ITS co-ordinates) when the user steps out of the building as GPS is recovered. The “jump” is the difference in the last dominant corrected PDR co-ordinates and the first recovered GPS co-ordinates. This is equal to the accumulated error of the PDR during the time spent by the user inside the building (i.e., when PDR corrections were not being updated instantaneously using the RTK-GPS).

	Walk 1	Walk 2	Walk 3	Walk 4
Last dominant PDR(Lat)	42.29406754	42.29469283	42.293688	42.29369639
Last dominant PDR(Long)	-83.71153664	-83.7114713	-83.71345745	-83.71349191
Recovered GPS (Lat)	42.29407585	42.29469192	42.29368167	42.29368364
Recovered GPS (Long)	-83.71152177	-83.711455	-83.71345969	-83.71347379
Jump (meter)	1.536	1.347	0.727	2.058

Table 3.1: Jumps in the ITS co-ordinates for short and simple walks

Short and complex walks: Relatively complex walks with duration between 3 and 5 minutes (indoors) are classified as short and complex walks. These walks involve relatively more turns, abrupt disturbances in motion, climbing, and sideward motion. Table 3.2 summarizes the "jumps" in the user's ITS position co-ordinates when the GPS is recovered.

	Walk 1	Walk 2	Walk 3	Walk 4
Last dominant PDR(Lat)	42.29369813	42.29369732	42.29370127	42.29369917
Last dominant PDR(Long)	-83.7134819	-83.71344807	-83.71345722	-83.7134601
Recovered GPS (Lat)	42.29368493	42.29367664	42.29367843	42.29367342
Recovered GPS (Long)	-83.71345852	-83.71346203	-83.71345254	-83.7134486
Jump (meter)	2.423	2.571	2.566	3.013

Table 3.2: Jumps in the ITS co-ordinates for short and complex walks

Longer walks: Relatively complex walks that last over 5 minutes (indoors) are classified as longer walks. These involve relatively more turns, abrupt disturbances in motion, climbing, and sideward motion. Table 3.3 summarizes the "jumps" in the user's ITS position co-ordinates when GPS is recovered.

	Walk 1	Walk 2	Walk 3	Walk 4
Last dominant PDR(Lat)	42.2936999	42.2936882	42.29393721	42.29484218
Last dominant PDR(Long)	-83.71349888	-83.7134695	-83.71318452	-83.71107857
Recovered GPS (Lat)	42.2936725	42.29367521	42.29390902	42.29483705
Recovered GPS (Long)	-83.71349173	-83.71345096	-83.71318044	-83.71103446
Jump (meter)	3.101	2.102	3.15	3.682

Table 3.3: Jumps in the ITS co-ordinates for longer walks

Sustainability test: To test the sustainability of the ITS we conducted a very long walk (over 30 minutes). The walk involved a lot of turns, abrupt disturbances in motion, climbing, and sideward motion in order to simulate a mobile user's natural motion in a complex environment. The walk was divided into 6 parts; 3 parts were of a short duration (less than 5 minutes indoors) and the other 3 were longer. At the end of each part, the user walked out of the building, recovered the RTK-GPS correcting the error in the ITS, and continued the walk into the building. Table 3.4 summarizes the results of the experiment used for testing ITS sustainability.

These experiments have helped conclude that this version of the ITS is very accurate for tracking smooth walks. The accuracy of the ITS reflects that of the PDR and degrades gracefully with both path complexity and time spent indoors. Once the accumulated drift in the ITS starts to exceed the satisfactory level, the user needs to step outdoors and recover the GPS signal to reset the corrections. Depending on the degree of accuracy required by the context-aware application, the required frequency of corrections can be determined. The average "jump" in the ITS co-ordinates when the GPS is recovered increases with the time spent indoors. This is expected because the corrections to the PDR are not being updated instantaneously due to RTK-GPS being unavailable. Table 3.5 summarizes the experimental results.

	Duration (Min:sec)	Last dominant PDR (Lat)	Last dominant PDR (Long)	Last dominant GPS (Lat)	Last dominant GPS (Long)	Jump (meter)
Part 1	4:30	42.29388927	-83.71325826	42.29388571	-83.71327255	1.24
Part 2	4:44	42.29389793	-83.71319148	42.2938827	-83.71321428	2.53
Part 3	4:52	42.29389745	-83.71328397	42.29389404	-83.7133059	1.85
Part 4	7:23	42.29390512	-83.71336043	42.29389401	-83.71340656	3.99
Part 5	8:18	42.29377712	-83.71348004	42.29374614	-83.71348369	3.45
Part 6	8:49	42.29378699	-83.71348071	42.29374324	-83.71347634	4.87

Table 3.4: Jumps in the ITS co-ordinates for the sustainability test walk

Type of walk	Average Duration Indoors	Average jump
Short and simple walks	3 minutes 45 seconds	1.4 meters
Short and complex walks	3 minutes 45 seconds	2.6 meters
Longer walks	6 minutes 15 seconds	3 meters

Table 3.5: Summary of the jumps in the ITS co-ordinates for different types of walks

The ITS jumps in the sustainability test walk are reflective of the average jump of several complex walks with similar duration, indicating that the ITS based on integrating RTK-GPS with the PDR system is sustainable. The ITS implementation and the experiments described in this section only correct the drift in the PDR system when RTK-GPS is available. To implement a positioning system where the drift accumulated by the PDR system is corrected even in a GPS-denied environment (for example indoors), the authors designed key improvements to the tracking system, as described in the following section.

3.4.2 Integrated Tracking System Based on Integrating Indoor Correction Points and PDR

This section describes the implementation of the ITS based on the integration of a data set of known location points called ‘correction points’ or ‘correction markers’ and the PDR system. The markers are made out of inexpensive rubber or wood panels with an imprinted numerical code, and can be assimilated into the architecture of the environment. The marker database serves as an infrastructure-based localization system, where location can be determined at discrete locations. The correction points are arranged as a grid in the desired environment and the accuracy of the localization system depends on the ‘compactness’ of the grid. The compactness of the grid, arranged for a specific context-aware computing application, depends on the desired tolerable values of the drift error accumulated in the integrated tracking system. The degradation curves of the ITS, as shown in Figures 9 through 12, illustrate how the accuracy of the PDR, and hence the integrated tracking system, changes with distance travelled and can be used to determine the compactness of the grid required for specific context-aware applications.

This version of the ITS continuously tracks the mobile user's PDR position, which keeps accumulating drift as the user continues navigating. Upon sensing a specific correction point, the ITS retrieves the 'true' location of the corresponding correction point from a pre-populated database. Using this correction point's position, the ITS then corrects the drift accumulated by the PDR system. This correction in drift is applied to the PDR location until the mobile user reaches another correction point and the ITS decides to correct the accumulated drift in position.

Thus, the error in the ITS would be reduced to the drift accumulated between the mobile user's current location and the last correction point at which the ITS chose to correct the drift instead of the drift accumulated during the entire walk, as would have been the case had the mobile user been tracked using only the PDR. As shown in Figure 3.6, the pre-determined correction points can help limit the drift accumulated by the PDR system in an indoor environment. Several validation experiments were performed to test the effectiveness of the ITS based on integrating the PDR system with a database of pre-determined correction points. Several points along the path of the mobile user during the experimental walks were marked on the floor (Points 0 through 24), as shown in Figure 3.8. The true locations of all such points were determined from the latest as-built floor plans and were stored in a database. Four specifically selected points (Points 4, 10, 15, and 20) at varying locations in the open loop trajectory of the experiments were selected to double as correction points.

A standardized format of Cartesian co-ordinates (x, y, z) along with time stamp markings were employed to denote the user's location. The ITS was modified to facilitate the evaluation of the drift along the x-axis and y-axis along with the Euclidean drift at each of the 25 points marked along the path of the mobile user during an experimental walk. The 16 experimental walks were divided into 4 sets of 4 walks, each set corresponding to a different correction marker at which the ITS decided to correct its drift.

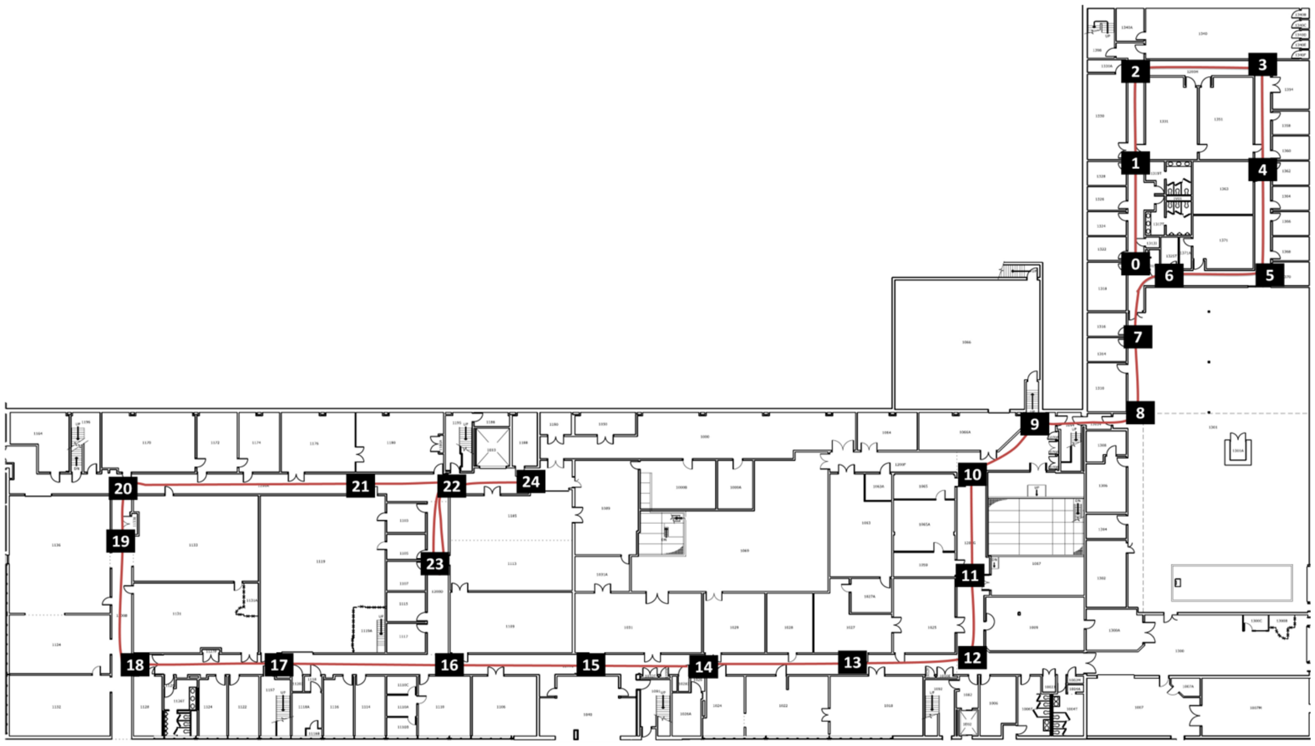


Figure 3.8: The layout of the specifically selected points (points 0 through 24) along the experimental walk path

Experimental walks with corrections made at point 4: Upon reaching point 4, the ITS retrieves the ‘true’ location of point 4 from the database to correct the drift accumulated by the PDR system. The degradation curves that depict the drift reflected in the ITS are presented by plotting the drift accumulated (along the x-axis, y-axis and Euclidean) against the distance travelled by the mobile user. The ITS degradation curves for 4 walks with drift corrections made at point 4 (50 m from the start) are illustrated in Figure 3.9.

Correction point at Point 4 (52 m from start of trajectory)

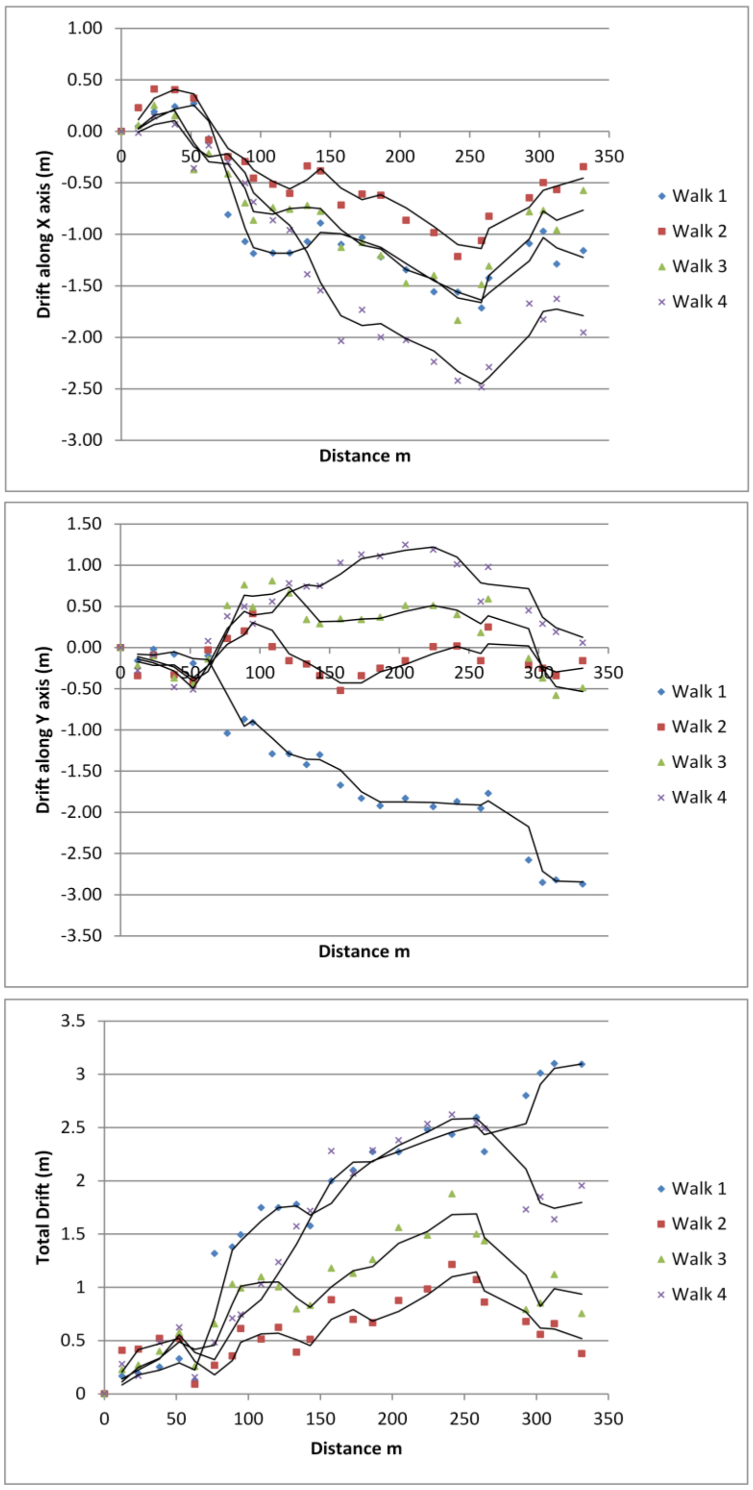


Figure 3.9: Degradation curves for ITS experimental walks with corrections made at point 4

Experimental walks with corrections made at point 10: The ITS degradation curves for the 4 walks with drift corrections made at point 10 (120 m from the start) are illustrated in Figure 3.10.

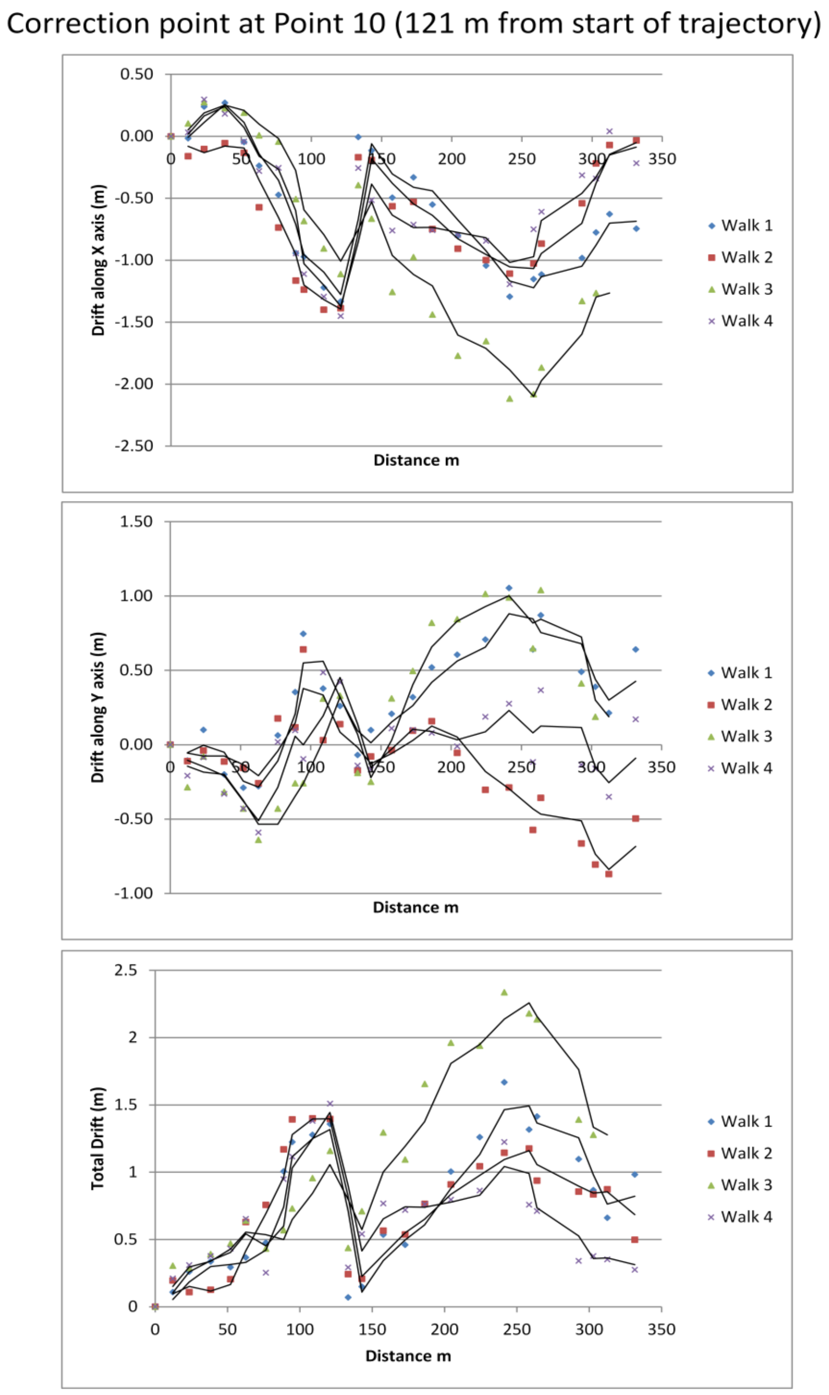


Figure 3.10: Degradation curves for ITS experimental walks with corrections made at point 10

Experimental walks with corrections made at point 15: The ITS degradation curves for the 4 walks with drift corrections made at point 15 (185 m from the start) are illustrated in Figure 3.11.

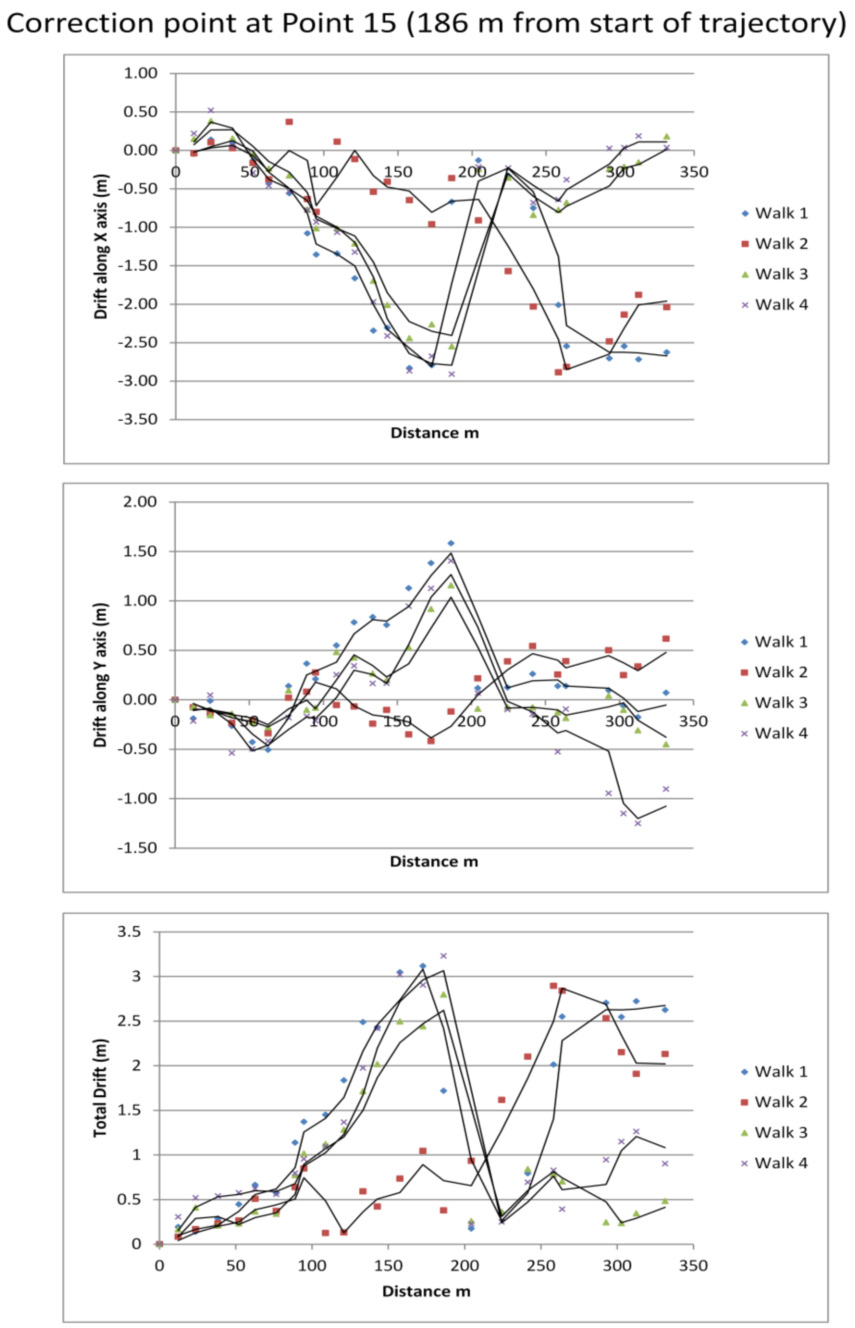


Figure 3.11: Degradation curves for ITS experimental walks with corrections made at point 15

Experimental walks with corrections made at point 20: The ITS degradation curves for the 4 walks with drift corrections made at point 20 (264 m from the start) are illustrated in Figure 3.12.

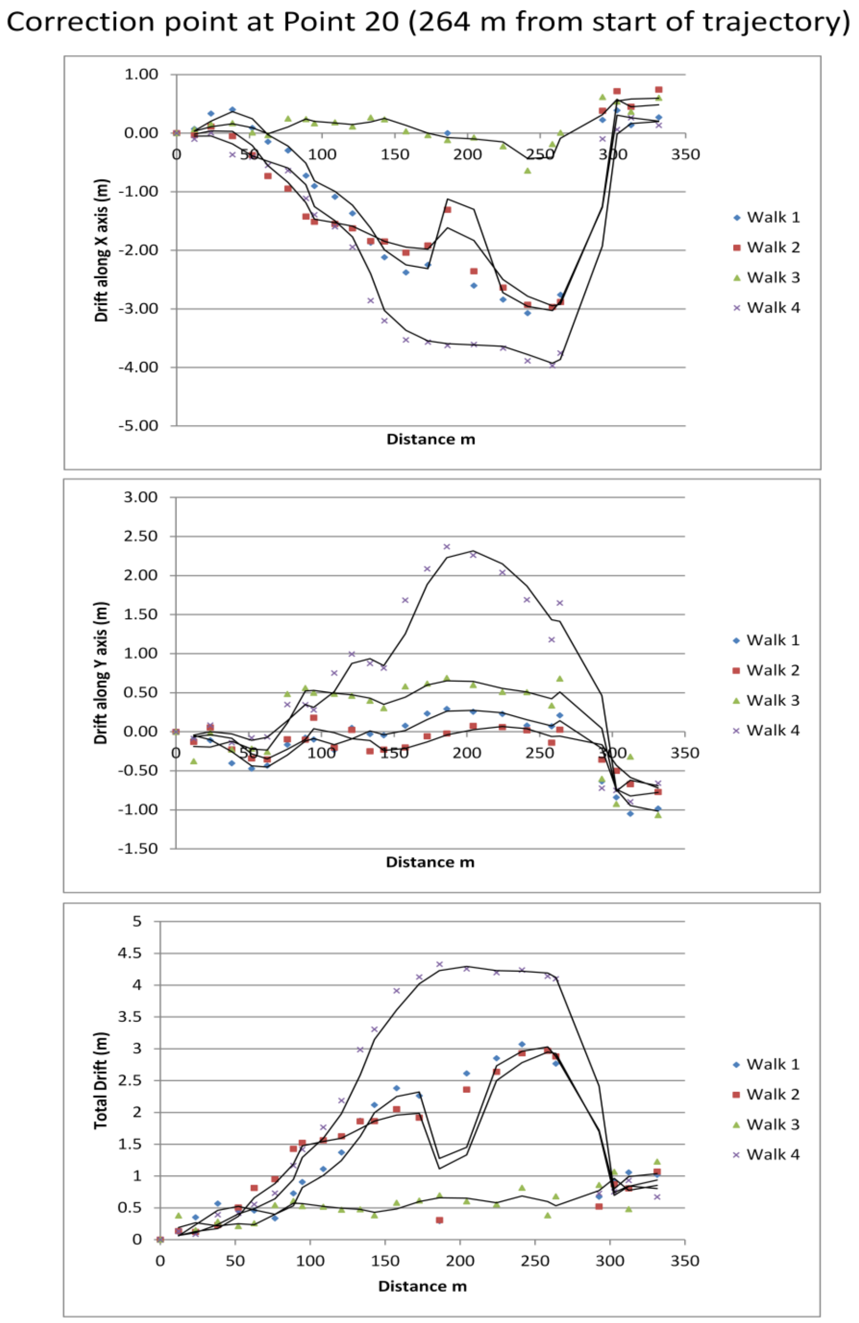


Figure 3.12: Degradation curves for ITS experimental walks with corrections made at point 20

The degradation curves in Figures 9 through 12 show the general trend that the drift in the ITS is being accumulated until the correction point, where the drift is corrected and set to zero. In the subsequent parts of the trajectory, the drift accumulated by the ITS is the drift accumulated by the PDR since encountering the latest correction point (and thus is not the entire drift accumulated from the start of the trajectory). The drift in the PDR, and consequently the ITS, can be accumulated in any direction and hence can sometimes result in cancelling previously accumulated drift even without any positional corrections. This phenomenon can be distinctly observed in the regression curves of the experimental walks conducted around the 240–260 meter region. This is likely caused by the change in the mobile user's direction—from walking along the x-axis to a brief walk along the y-axis for 20 m (at Point 18) and then reverting back to walking along the x-axis (but in the opposite direction) around the 240 m mark (at Point 20).

During a walk, the previously accumulated drift in the PDR system may be canceled later on even without any ITS corrections but such a 'correction' is not a given. However, the maximum drift that is accumulated by the PDR system, and therefore reflected in the integrated tracking system between consecutive corrections, is limited as a linear function of the distance travelled. As mentioned previously, the compactness of the correction points documented in the database, for this version of the ITS, would depend on the degree of accuracy desired by the mobile user performing the specific task. When implemented on a large scale, this version of the ITS would require relatively little fingerprinting and significantly lower computational power compared to traditional indoor positioning systems like WLAN, UWB, and indoor GPS-based positioning systems. Based on the regression curves plotted, alarm systems can be designed for the ITS that will warn the ITS and the user to trigger a correction when the system expects the accumulated drift to approach intolerable values. However, the main drawback to the drift corrections in PDR system through the version of the ITS presented in this section is that the corrections can only be made at specific pre-selected points in the indoor environment.

Moreover, this version of the ITS requires the indoor environment to be set up with correction markers at known locations. To overcome these drawbacks, further investigation was done to develop a hybrid tracking system that corrects the drift accumulated in the PDR system by complementing the PDR system with human intelligence and the mobile user’s knowledge and discernment of the environment. The design, implementation and validation of the hybrid tracking system based on the PDR system and human intelligence is described in the following section. The author is also investigating the applicability of using natural and fiduciary markers as correction points and computer vision techniques to recognize them in order to further minimize human effort in correcting the accumulated drift in the ITS.

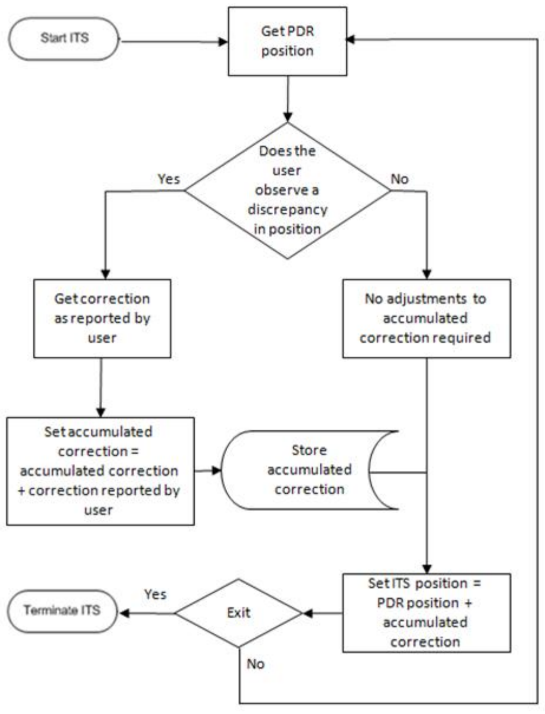
3.4.3 Human Intelligence-Based Tracking System

The main drawback of the ITS described in the previous sections of this chapter is that the mobile user is forced to carry a relatively large payload in the form of receivers. Moreover, if the ITS does not encounter any corrector points or infrastructure-based localization systems for a relatively long duration, the drift accumulated in the PDR might exceed tolerable limits. This section presents an ITS based on establishing bi-directional communication between the mobile user and the PDR system. By using the knowledge of the mobile user, based on the observation of the environment, the ITS can effectively correct large amounts of drift accumulated in the PDR.

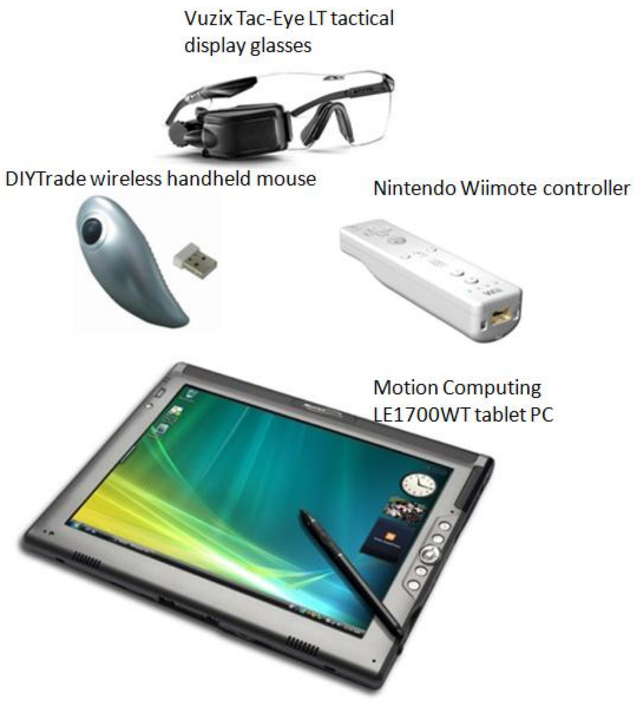
This version of the ITS is particularly suitable in applications where localization is the primary purpose of the system. Although the ITS algorithm provides the ability for a human user to correct the position, it may not be desirable to burden the user with position correction activity while performing critical tasks (for example, while looking for survivors in an emergency response scenario). The ITS consists of a PDR system that is connected to the computation center—a tablet computer—where the ITS software is installed. The algorithm of the ITS is shown in Figure 3.13(a) and provides methods for users to visualize and correct their position when desired and feasible. The visualization-based correction application is installed in the aforementioned computation center and does not increase the payload of the user.

The mobile user can visualize the motion trajectory in a virtual environment where the user's avatar is shown as a red dot against the backdrop of the floor plans representing the currently occupied area. As the drift error is accumulated in the PDR, the mobile user's virtual location continues to drift away from the true location. The mobile user visually observes the true location relative to the real-world environment and compares it to the PDR location represented by the location of the virtual avatar relative to the backdrop. If the mobile user visually observes a significant discrepancy between the true location and the sensed (and depicted) virtual location, the discrepancy is communicated to the tracking system by adjusting the user's avatar on the computer screen so that it represents the user's estimate of current location relative to the surroundings (e.g., walls, doors, corridors, etc.).

As mentioned previously, the discrepancy in location is communicated through the visualization based correction application on the tablet computer. If the task requires the use of a non-tablet computation center, for example a belt-mounted computer, then the application can be visualized through the use of tactical display glasses, shown in Figure 3.13(b). Tactical display glasses are typically used in augmented reality applications to reduce payload and enhance usability. The communication is facilitated by connecting the tactical display glasses to a controller—similar to a handheld mouse or a Nintendo Wii nunchuck controller—to move the avatar in the 2D or 3D environment while keeping at least one hand of the user free for performing the tasks that the context-aware application was designed for.


Start ITS
Get PDR position
Does the user observe a discrepancy in position
Yes
No
Get correction as reported by user
No adjustments to accumulated correction required
Set accumulated correction = accumulated correction + correction reported by user
Store accumulated correction
Set ITS position = PDR position + accumulated correction
Exit
Yes
No
Terminate ITS

(a)


Vuzix Tac-Eye LT tactical display glasses
DIYTrade wireless handheld mouse
Nintendo Wiimote controller
Motion Computing LE1700WT tablet PC

(b)

Figure 3.13: (a) Algorithm to integrate PDR system with human intelligence-based corrections and (b) devices used to implement the discrepancy correction application

Once the mobile user satisfactorily adjusts the location, navigation can be continued and the adjustment made to the PDR position is applied as a correction to the subsequent part of the walk. The elegance of this tracking system lies in the fact that the corrections to the PDR position can be updated several times at any location in the trajectory of the user's walk, and that this requires no preinstalled infrastructure whatsoever. For this version of the ITS to work optimally, the users need to have some familiarity with the environment or the environment needs to be delineated with appropriate and adequate signs, for example—door numbers or stairwells—for the users to know where they are at a given time.

The validation experimental set up was similar to the setup described in Section 3.4.2; it is shown in Figure 3.8. However, in this case, there were no specific pre-determined correction points. The room numbers and corridors were employed by the mobile user to know where s/he was at any given time. The hybrid tracking system had the facility to update the drift correction wherever and whenever the mobile user chose to do so by visually estimating the depicted avatar (the red dot) in the virtual environment relative to true position, as illustrated in Figure 3.14.

The drift (along x-axis, y-axis and Euclidean drift) data collected for all 4 walks at each of the 25 points (Points 0 through 24) is plotted against the distance travelled by the mobile user to obtain the respective degradation curves for the hybrid tracking system, as shown in Figure 3.15. The uncorrected drift in the PDR system plotted in Figure 3.15 is the average drift accumulated in the PDR for the 16 experimental walks.

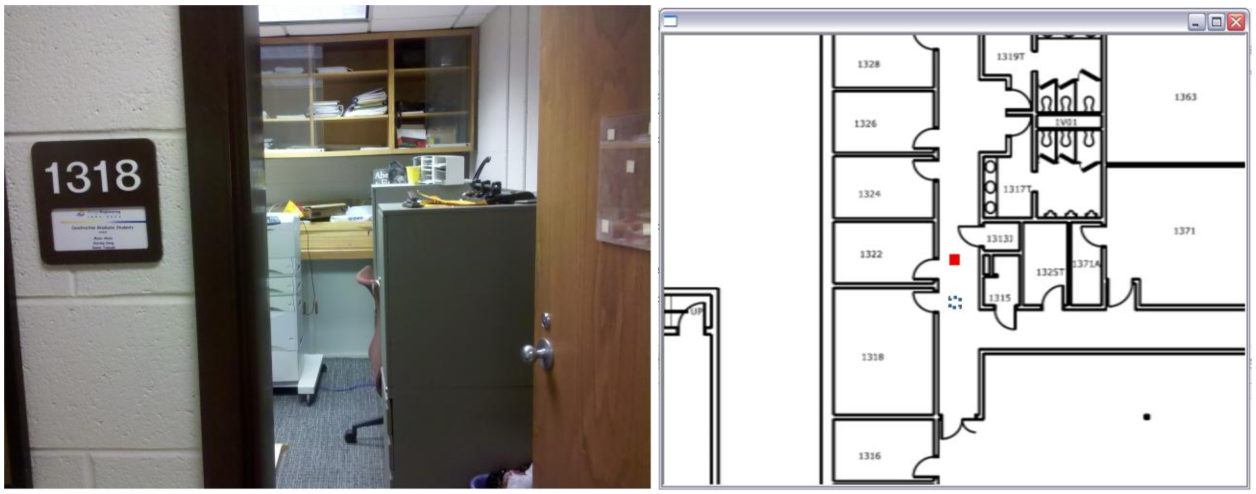


Figure 3.14: The first person view of the mobile user’s environment (left) and the virtual representation of the user’s sensed location with the floor plans used as the environment backdrop (right)

During the course of the four experimental walks, the user was allowed to correct location drift as often as desired, depending on when a discrepancy in position was observed (i.e., based on their individual tolerance for observed location discrepancy). Walks 1 and 2 were performed by one subject, while walks 3 and 4 were performed by another subject; both subjects were unfamiliar with the environment. However, as noted earlier, the subjects employed room numbers placed at the door of each room, as well as the floor plan overlay, to know where they were at any given time. The reduction of drift is dependent on the subject’s tolerance of error, fatigue, familiarity with the environment, and ability to intelligently employ architectural features of the environment as signs to know where their location is at any given time.

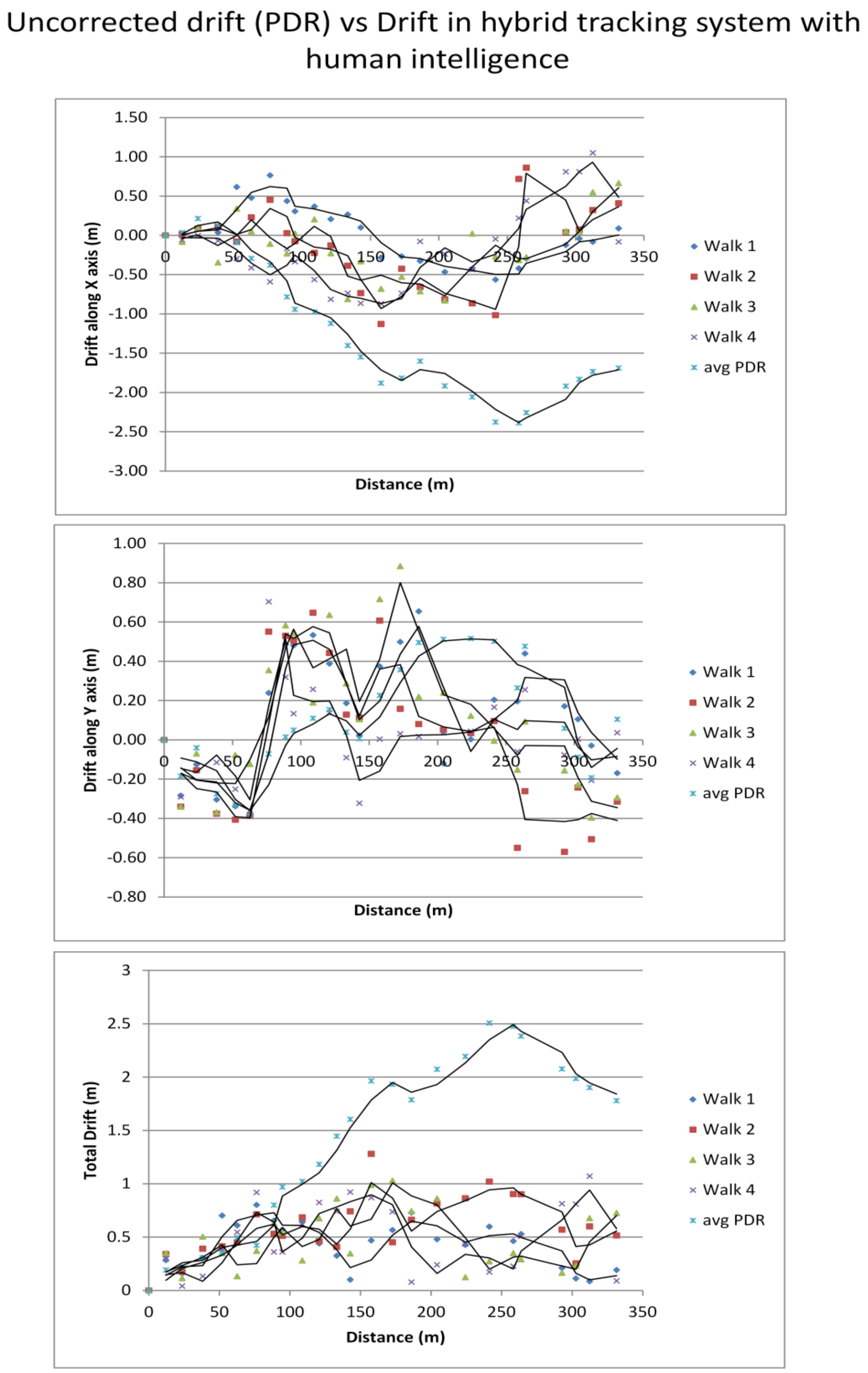


Figure 3.15: Degradation curves for the ITS based on PDR and human intelligence compared with the degradation of the PDR alone

The tracking system based on human intelligence and intervention degrades more gracefully than the ITS based on integrating the correction points database and PDR system. This fact is especially highlighted in the degradation curves that plot the Euclidean drift values against the distance travelled by the mobile user. The degradation curves also suggest that human intelligence can identify the discrepancy in true location and virtual location even when the difference is less than 1 m by comparing the real environment with the virtual representation. The human intelligence-based tracking system provides a practical alternative especially when the task being performed by the mobile user is lengthy, tiresome, and sensitive to the payload being carried. While performing tasks such as bridge inspections or post-disaster assessments of damaged buildings, PDR-based tracking systems integrated with human intelligence offer significant promise in continually identifying the spatial context of the mobile user.

3.5 Summary and Conclusions

This chapter presented the overarching architecture and algorithm of the ITS for integrating infrastructure-based localization systems and infrastructure-independent localization technologies for the ubiquitous tracking of a mobile user. A version of the ITS based on integrating RTK-GPS and PDR was developed successfully, and several validation experiments were conducted for varying complexities of the mobile user's trajectory, proving the sustainability of the ITS. This version of the ITS was found to be very accurate for tracking smooth walks, however the accuracy degrades with extreme legged motion. The accuracy of the ITS was reflective of the PDR and was within 0.5 m for every minute spent in a GPS denied environment. However, the localization drift error would exceed tolerable limits upon spending a considerable portion of the walk indoors.

In order to improve the accuracy in tracking a mobile user while indoors, the authors further developed and implemented a version of the ITS based on integrating a database of pre-determined known correction point locations and PDR. To test this version of the ITS, several validation experiments were conducted in the G.G. Brown building at the University of Michigan. The obtained results and degradation curves demonstrated that while the drift error was reduced by the correction points, the reduction in drift error was largely dependent on the density of the correction points (i.e., the proximity of correction points to one another) present in the pertinent environment. When the correction points are arranged such that the mobile user encounters at least 1 correction point for every 300 m of his/her walk, it is observed that the positional error rarely exceeds 4 m. The disadvantage of this version of the ITS is that it requires the indoor environment to be set up with correction points apriori. Moreover, if the mobile user does not encounter any correction points along the walk for a significant distance, the drift error will exceed tolerable limits.

As a further improvement, a hybrid tracking system based on PDR and human intelligence was developed and validated. To test the hybrid tracking system, several validation experiments were conducted in the G.G. Brown building at the University of Michigan. The obtained results and degradation curves demonstrated that the hybrid tracking system was sustainable and the drift error was limited to within 1 m irrespective of the distance travelled by the mobile user. The accuracy, strengths, and weakness of the aforementioned tracking systems are summarized in Table 3.6. The results of these experiments highlighted the potential of using the ITS in general, and the hybrid tracking based on human intelligence in particular, to continuously and accurately track a mobile user for ubiquitous context-aware engineering applications with sub-meter level accuracy.

Positioning System	Accuracy	Weakness	Strength
RTK-GPS (Garmin A-GPS)	2.5 cm to 5 cm horizontally, 5 cm to 7.5 cm vertically	Requires a clear line of sight to at least 4 satellites, has a heavy payload, is quite expensive	Has a high accuracy, environment does not need to be retrofitted with infrastructure
WLAN (Ekahau Positioning System)	1.5 m to 2 m	Requires pre-installed infrastructure and calibration, infrastructure may be partially or fully damaged in the case of disaster	Line of sight is not required, is easy to deploy and is economical
UWB (Sapphire Dart UWB Digital Active Real Time Tracking System)	0 cm to 50 cm	Infrastructure is expensive to install, infrastructure may be partially or fully damaged in the case of disaster, requires a direct line of sight	Deployment is easy as calibration is not required
PDR System (Personal Odometry System)	Less than 2% of total distance walked	The drift error is accumulated continuously and exceeds tolerable values when enough distance is covered, accuracy degrades with distance and extreme modes of legged locomotion	Is independent of installed infrastructure, has a light payload, is inexpensive, does not require replacement in the case of disaster
Integrated Tracking System based on RTK-GPS and PDR	When RTK-GPS is available: 2.5 cm to 5 cm horizontally and 5 cm to 7.5 cm vertically; When RTK-GPS is unavailable: less than 2% of latest non-GPS distance walked	Has a heavy payload, is quite expensive, drift error may exceed tolerable values if RTK-GPS corrections are not made frequently (i.e. non-GPS walk distance is considerably large)	Environment does not need to be retrofitted with infrastructure, system works even when line of sight to satellites is lost temporarily
Integrated Tracking System based on Indoor Correction Points and PDR	Error due to accumulated drift can be limited within tolerable values based on the 'compactness' of the correction points grid.	Tracking system may be unreliable in emergency scenarios, environment must be retrofitted with correction marker grid so that the markers are frequently encountered	Infrastructure is inexpensive and is easily replaceable if damaged, has a light payload, drift error can be limited within tolerable values
Human Intelligence-based Tracking System	Less than 1 m	Needs familiarity with environment, has the ability to adopt architectural features as appropriate reference points, accuracy is dependent on an individual's tolerance of error	Environment does not need to be retrofitted with infrastructure, has a light payload

Table 3.6: Comparative summary of the three versions of ITS developed with RTK-GPS, WLAN-based, UWB-based, and PDR-based localization systems

3.6 Acknowledgments

The presented work is partially supported by the National Science Foundation (NSF) through grant CMMI-0927475. The author gratefully acknowledges the NSF's support. The author would also like to thank PhD students Sanat Talmaki, Chunxia Li, and Chen Feng for their assistance in conducting some of the experiments. Any opinions, findings, conclusions, and recommendations expressed in this chapter are those of the author and do not necessarily reflect the views of the NSF, or the individuals mentioned here.

3.7 References

Aittola, M., Ryhänen, T., and Ojala, T. (2003). “SmartLibrary—Location-Aware Mobile Library Service,” Proc. of the 5th Int. Symp. on Human Computer Interaction with Mobile Devices and Services, Udine, Italy, pp. 411–416.

Akula, M., Dong, S., Kamat, V.R., Ojeda, L., Borrell, A., and Borenstein, J. (2011). "Integration of Infrastructure-Based Positioning Systems and Inertial Navigation for Ubiquitous Context-Aware Engineering Applications,” Advanced Engineering Informatics, Vol. 25(4), ICCCBE 2010 Special Issue on Computing in Civil and Building Engineering, Elsevier Science, New York, NY, pp. 640–655.

Aziz, Z., Anumba, C.J., Ruikar, D., Carrillo, P.M., and Bouchlaghem, D.N. (2005). “Context Aware Information Delivery for On-Site Construction Operations,” Proc. of the 22nd CIB-W78 Conference on Information Technology in Construction, Institute for Construction Informatics, Technische Universitat Dresden, Germany, CBI Publication No: 304, pp. 321–32.

Aziz, Z., Anumba, C.J., Ruikar, D., Carrillo, P.M., and Bouchlaghem, D.N. (2006). "Intelligent Wireless Web Services for Construction – A review of Enabling Technologies," Automation in Construction, Elsevier, Vol. 15(2), 113–23.

Beauregard, S. and Haas, H. (2006). "Pedestrian Dead Reckoning: A Basis for Personal Positioning," Proc. of the 3rd Workshop on Positioning, Navigation, and Communication, Hannover, Germany, pp. 27–36.

Bowden, S.L., Dorr, A., Thorpe, A., and Anumba, C.J. (2004). "Mapping Site Processes for the Introduction of Mobile IT," Proc. of the ECPPM eWork and eBusiness in Architecture, Engineering and Construction, AA Balkeman Publishers, pp. 491–498.

Buick, R. (2006). "RTK Base Station Networks Driving Adoption of GPS +/- 1 inch Automated Steering among Crop Growers White Paper," Trimble Navigation Limited Website <http://www.trimble.com/pdf/AG_RTK%20BSNetworks_WP_0806.pdf> as accessed on 4th April, 2013.

Burrell, J. and Gay, K. (2001). "Collectively Defining Context in a Mobile, Networked Computing Environment," Proc. of the Conference on Human Factors in Computing Systems, Association for Computing Machinery (ACM), New York, NY, pp. 231–232.

Cobb, H.S. (1997). “GPS Pseudolites: Theory, Design and Applications,” PhD Dissertation (SUDAAR 707), Stanford University, Stanford, CA.

Collin, J. (2003). “Indoor Positioning System Using Accelerometry and High Accuracy Heading Sensors,” Proc. of GPS/GNSS 2003 Conference (Session C3), Portland, OR, pp. 1–7.

Cox, S., Perdomo, J., and Thabet, W. (2002). “Construction Field Data Inspection Using Pocket PC Technology,” Proc. of the Int. Council for Research and Innovation in Building and Construction, Aarhus, Denmark, pp. 243–251.

Davies, N., Cheverst, K., Mitchell, K., and Friday, A. (1999). “Caches in the Air: Disseminating Information in the Guide System,” Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), Institute of Electrical and Electronics Engineers (IEEE), New Orleans, Louisiana, pp. 11–19.

Dissanayake, G., Sukkarieh, S., Nebot, E., and Durrant-Whyte, H. (2001). “The Aiding of Low-Cost Strapdown Inertial Measurement Unit using Vehicle Model Constraints for Land Vehicle Applications,” IEEE Transactions on Robotics and Automation, Institute of Electrical and Electronics Engineers (IEEE), Vol. 17(5), pp. 731–747.

Fang, L., Antsaklis, P.J., Montestruque, L.A., McMickell, M.B., Lemmon, M., Sun, Y. Fang, H., Koutroulis, I., Haenggi, M., Xie, M., and Xie, X. (2005). “Design of a Wireless Assisted Pedestrian Dead Reckoning System - the NavMote Experience,” IEEE Transactions on

Instrumentation and Measurement, Institute of Electrical and Electronics Engineers (IEEE), Vol. 54(6), pp. 2342–2358.

Farrar, M.M. (2011). "The AASHTO Manual for Bridge Evaluation," American Association of State Highway and Transportation Officials (AASHTO), AASHTO Bookstore, Edition 2, ISBN 1-56051-496-1, pp. 1–574.

Fischmeister, S., Menkhaus, G., and Pree, W. (2002). "MUSA-Shadows: Concepts, Implementation, and Sample Applications; a Location-Based Service Supporting Multiple Devices," Proc. of the 40th Int. Conference on Technology of Object-Oriented Languages and Systems (TOOLS), Sydney, Australia, pp. 71–79.

Fleck, M., Frid, M., Kindberg, T., O'Brien-Strain, E., Rajani, R., and Spasojevic, M. (2002). "From Informing to Remembering - Deploying Ubiquitous Systems in Interactive Museums," IEEE Pervasive Computing, Institute of Electrical and Electronics Engineers (IEEE), Vol. 1(2), pp. 13–21.

Forssell, U., Hall, P., Ahlqvist, S., and Gustafsson, F. (2002). "Novel Map-Aided Positioning System," Proc. of FISITA 2002 World Automotive Congress, International Federation of Automotive Engineering Societies (FISITA), Helsinki, Finland.

Hazas, M., and Hopper, A. (2006). "Broadband Ultrasonic Location Systems for Improved Indoor Positioning," IEEE Transactions on Mobile Computing, Institute of Electrical and Electronics Engineers (IEEE), Vol. 5, pp. 536–547.

Huang, C. Liao, Z., and Zhao, L. (2010). "Synergism of INS and PDR in Self-Contained Pedestrian Tracking with a Miniature Sensor Module," IEEE Sensors Journal, Institute of Electrical and Electronics Engineers (IEEE), Vol. 10(8), pp. 1349–1359.

Khoury, H.M., and Kamat, V.R. (2008). "Indoor User Localization for Rapid Information Access and Retrieval on Construction Sites," Proc. of the 15th Annual Workshop of the European Group for Intelligent Computing in Engineering, European Group for Intelligent Computing in Engineering (EG-ICE), Plymouth, UK, pp. 497–507.

Khoury, H.M., and Kamat, V.R. (2009). "High-Precision Identification of Contextual Information in Location-Aware Engineering Applications," Advanced Engineering Informatics, Elsevier Lt., Vol. 23, pp. 483–496.

Laukkanen, M., Helin, H., and Laamanen, H. (2002). "Tourists on the Move," Cooperative Information Agents VI - 6th International Workshop, Lecture Notes in Computer Science, Springer, Germany, pp. 36–50.

Lemmon, T., and Wetherbee, L. (2005). ‘Trimble Integrated Surveying Techniques White Paper,” Trimble Navigation Website <http://trl.trimble.com/docushare/dsweb/Get/Document-239290/022543-133_Integrated_Surveying_WP_0605.pdf>, as accessed on 3rd April, 2013.

Mautz, R. (2009). “Overview of Current Indoor Positioning Systems,” Geodezija Ir Kartografija / Geodesy and Cartography, Vol. 35(1), pp. 18–22.

May, A., Mitchell, V., Bowden, S., and Thorpe, T. (2005). “Opportunities and Challenges for Location Aware Computing in the Construction Industry,” Proc. of the Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI’05), Association for Computing Machinery (ACM), ACM Press, Austria, pp. 255–258.

Meng, X., Yang, L., Aponte, J., Hill, C., Moore, T., and Dodson, A.H. (2008). “Development of Satellite Based Positioning and Navigation Facilities for Precise ITS Applications,” Proc. of the 11th International IEEE Conference on Intelligent Transportation Systems, Institute of Electrical and Electronics Engineers (IEEE), Beijing, China, pp. 962–967.

Ojeda, L., and Borenstein, J. (2007). “Personal Dead Reckoning System for GPS-denied Environments,” IEEE International Workshop on Safety, Security and Rescue Robotics, Institute of Electrical and Electronics Engineers (IEEE), Rome, Italy.

Pradhan, A., Ergen, E., and Akinci, B. (2009). "Technological Assessment of Radio Frequency Identification Technology for Indoor Localization," Journal of Computing in Civil Engineering, American Society of Civil Engineers (ASCE), Vol. 23(4), pp. 230–238.

Roberts, G.W., Dodson, A.H., and Ashkenazi, A. (2000). "Experimental Plant Guidance and Control by Kinematic GPS," Proc. of the Institution of Civil Engineers – Civil Engineering, Institution of Civil Engineering (ICE) Virtual Library, Vol. 138(1), pp. 19–25.

Roberts, G.W., Ogundipe, O., and Dodson, A.H. (2002). "Construction Plant Control Using RTK GPS," Proc. of the XXII Int. Congress of the FIG, TS 6.2 Engineering Surveys for Construction Works and Structural Engineering II, Washington DC, USA, pp. 1–13.

Singhvi, V., Fish, W., and Terk, M. (2003). "Context-Aware Information System for Construction Site Applications," Proc. of the ASCE Construction Research Congress- Winds of Change: Integration and Innovation in Construction, American Society of Civil Engineers (ASCE), Honolulu, Hawaii, pp. 981–988.

Skattor, B. (2007). "Design of Mobile Services Supporting Knowledge Processes on Building Sites," Proc. of the World Congress on the Management of eBusiness (WCMeB), Institute of Electrical and Electronics Engineers (IEEE), pp. 10–17.

Torres-Solis, J., and Chau, T. (2010). "Wearable Indoor Pedestrian Dead Reckoning System," Pervasive and Mobile Computing, Elsevier, Vol. 6(3), pp. 351–361.

Chapter 4

Controlled Drilling for Embeds into Reinforced Concrete Bridge Decks

4.1 Introduction

For construction workers, civil engineering personnel, and industrial workers, the ability to perceive, comprehend, and analyze the spatial context of their work environment is imperative to complete their tasks competently, efficiently, and safely. Compared to semi-structured workspaces such as industrial, manufacturing, and assembly line worksites, construction worksites are dynamic, unstructured, and continuously evolving workspaces that result in unique characteristics (Son et al 2008). Civil engineering projects, especially in urban areas, are characterized by narrow, constrained workspaces leading to the limited visibility of resources resulting in an increase in the probability of collisions between equipment, workers, materials, and infrastructure (Teizer et al 2010; Cheng and Teizer 2011).

Projects involving excavation and drilling increase the risk of collisions among equipment and worksite resources due to the occluded vision of operating personnel (Saidi et al 2011). Moreover, large equipment operators are faced with the additional challenge of overcoming blind spots on equipment and haul roads (Talmaki et al 2010). Additionally, certain projects—such as two cranes working in tandem to lift heavy objects—inherently pose constraints on operators' ability to comprehend their spatial context, and so challenge their ability to analyze their work environment in order to achieve their objectives (Zhang et al 2008).

Drilling personnel performing drilling operations to place embeds into reinforced concrete decks are faced with the risk of striking rebar and buried utility lines. As mentioned above, another example of an occlusion hampering operator visibility, and therefore of operation efficiency, is the case of excavation. Excavation operations in the presence of underground utility lines face the constant risk of striking buried utilities, resulting in significant damage to property, injuries, and fatalities. In the absence of equipment and infrastructure tracking, operators must rely on planning, judgment, and experience to estimate the areas that are safe for drilling and excavation. Therefore, applications that provide operators with information regarding their spatial context would enhance their decision-making accuracy and their ability to perform operations safely and with increased levels of efficiency.

Such context-aware computing applications periodically examine and react to changing context. Environmental variables that are typically used to communicate contextual information usually typically include, but are not limited to, location (where), identity (who), time (when), and activity (what) (Burrell and Gay 2001). Context-aware computing applications are implemented using a mobile, distributed computing system—a collection of mobile and stationary sensing and computing devices that are cooperating and communicating on the targeted user's behalf (Schilit et al 1994).

A framework to develop real-time monitoring systems for civil engineering applications is presented in this chapter. Next, the author presents a motivating scenario and the technical approach, based on the developed framework, toward solving the scenario-specific problems. The author then presents methodologies used to develop context-aware applications that address the scenario. The research that investigated and evaluated the performance of the developed methodologies in comparison with the ground truth is also presented.

4.1.1 Importance of the Research

The construction industry continues to have a high number of accidents and a poor record of safety compared to other industries, despite recent significant efforts to improve safety. The U.S. Bureau of Labor Statistics estimated that the construction industry historically has had the highest total number of fatalities in all industries, as shown in Table 4.1 (BLS 2010).

	Construction	Protective Services	Farming, Fishing, and Forestry	Manufacturing
2010	780	261	276	363
2009	838	244	239	326
2008	977	306	286	354
2007	1,172	346	258	380
2006	1,273	284	297	423

Table 4.1: Historic chart for number of fatalities per occupation

Furthermore, the rate of fatality in the construction industry is also relatively high. In 2010, the fatality rate for the construction industry was 9.8 per 100,000 employees—the fourth highest across all industry sectors, as shown in Figure 4.1 (BLS 2010).

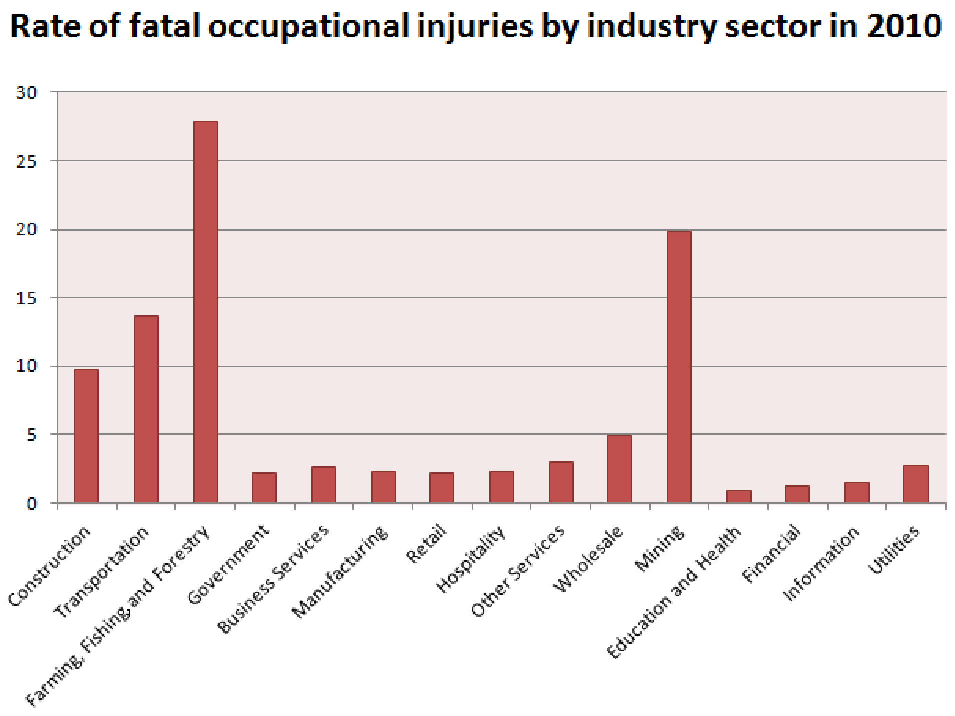


Figure 4.1: Fatality rates by occupation in the year 2010 per 100,000 employees

These fatalities have been ascribed to several contributing factors in order to better understand the factors contributing to these fatalities and to effectively implement safety management practices. The National Institute for Occupational Safety and Health (NIOSH) classifies the contributing factors into the following categories: lack of hazard recognition, lack of coordination of work tasks, worker inexperience, deviation from standard operating procedure, fast-track scheduling, and employers' lack of written task-specific work procedures (NIOSH 2007). Real-time monitoring systems usually address mistakes caused due to a lack of hazard recognition by using context-aware computing techniques.

Lack of hazard recognition occurs either due to the failure to perceive a potentially hazardous scenario or due to the failure in interpreting the perceived scenario as hazardous. As mentioned previously, urban construction projects are characterized by narrow, constrained workspaces leading to the limited visibility of resources (Teizer et al 2010; Cheng and Teizer 2011), resulting in an increase in the probability of failing to perceive and identify hazardous scenarios as such.

A significant percentage of highway construction takes place during night time when the traffic flow is minimal (Arditi et al 2005; Hinze and Teizer 2011). In such projects, the lack of visibility becomes a major contributing factor for workplace accidents. Poor visibility and lack of lighting was found to be a leading cause for accidents involving workers colliding with traffic and construction equipment (Arditi et al 2005). Blind spots and obstructions were found to account for nearly 75% of visibility-related fatalities in construction (Hinze and Teizer 2011).

Certain projects—the ones that involve concealed or buried infrastructure—are inherently fraught with problems concerning limited visibility and occlusion. Drilling, excavation, and trenching operations result in frequent accidents when equipment strikes the concealed infrastructure; these operations account for a significant amount of contingency cost to compensate injured personnel and damaged property. Context-sensing technology and/or computer vision techniques are employed to obtain environmental variables that complement and enhance personnel perception in order to identify hazards.

Localization and tracking technologies have been used to capture changing worksite information to develop real-time monitoring systems to warn workers of danger (Abderrahim et al 2005;

Chae and Yoshida 2010; Fullerton et al 2009; Ruff 2008; Talmaki et al 2010; Teizer et al 2010; Wu et al 2010). Real-time monitoring systems track the resource’s (worker, equipment, objects) spatial-context variables, such as location and orientation, in real-time using localization and tracking technologies. These systems then automatically compare the identified spatial-context with predefined scenarios identified as being dangerous. Such monitoring systems help prevent accidents by alerting personnel to potential dangers.

4.1.2 Real-Time Monitoring

The implementation of real-time monitoring systems based on context-aware computing transcends the reality–virtuality continuum as shown in Figure 4.2. Immutable environment variables (such as fixed infrastructure) and constraints (such as hazard scenarios) are captured and stored as pre-processed data. Dynamic environmental variables and characteristics are continuously captured in real-time through appropriate sensing technologies, and are processed and interpreted in the virtual environment. The relevant pre-processed and real-time data are then passed onto the hazard detection algorithm that determines whether the captured scenario is hazardous or not. The results, including any potential warnings, are communicated to the appropriate personnel through audio-visual methodologies that can fall anywhere on the reality–virtuality continuum. Based on any potential warnings communicated, the operator makes the decision whether to continue with the scenario or seek an alternative.

The monitoring system must be designed such that the appropriate data are captured, processed, and analyzed, such that warnings are issued to the personnel, if necessary, within the duration in which the current scenario could transform into the potentially hazardous scenario that has been identified. The monitoring system is designed to be "real-time" by ensuring that the context-aware computing algorithms allow for appropriate safety factors and tolerance variables. The remainder of this chapter explains how the developed framework can be utilized to design and implement real-time warning systems based on context-aware computing for a particular motivating scenario that is described in the following section.

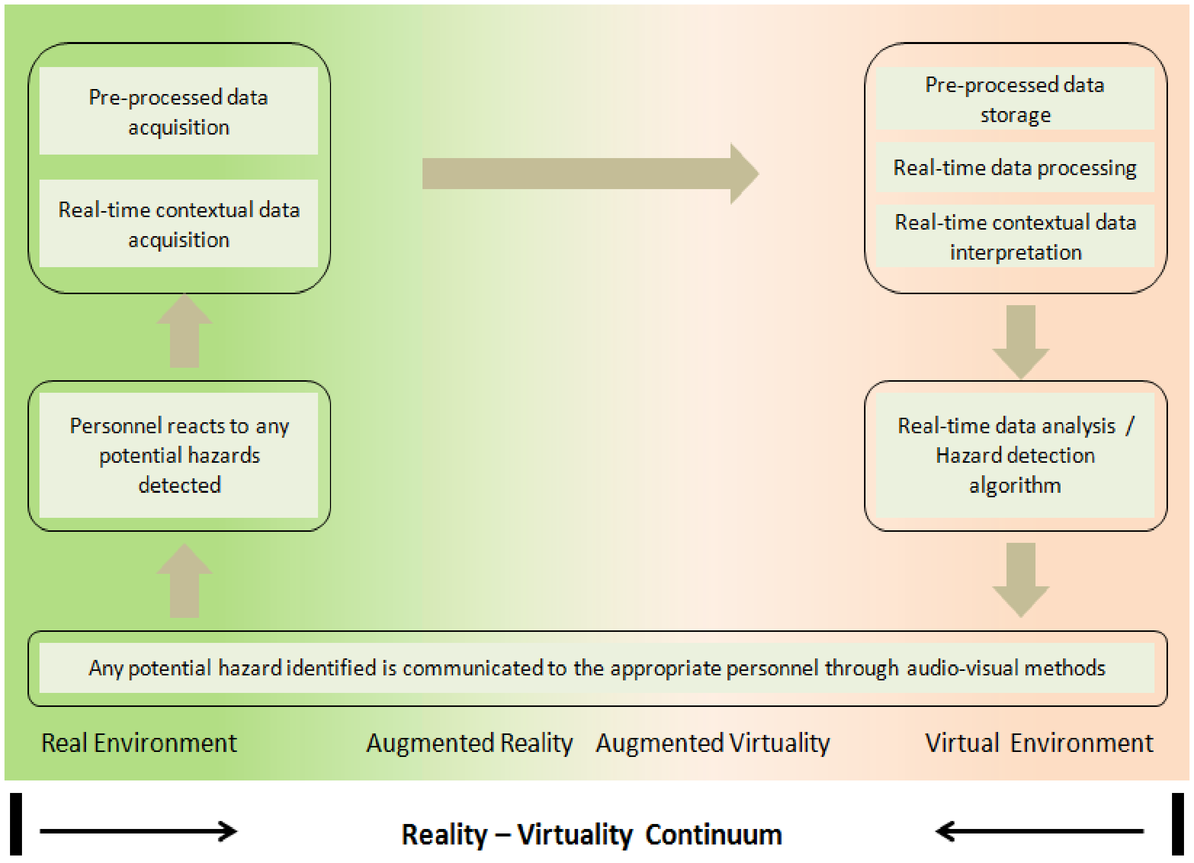

Pre-processed data acquisition
Real-time contextual data acquisition
Personnel reacts to any potential hazards detected
Pre-processed data storage
Real-time data processing
Real-time contextual data interpretation
Real-time data analysis / Hazard detection algorithm
Any potential hazard identified is communicated to the appropriate personnel through audio-visual methods
Real Environment
Augmented Reality
Augmented Virtuality
Virtual Environment
Reality – Virtuality Continuum

Figure 4.2: Real-time monitoring systems within the reality–virtuality continuum

4.2 Motivating Scenario

The construction of a railway line requires placing embeds into reinforced concrete decks along the length of the railway line. While drilling into reinforced structures, it is of the utmost importance that the drill bit avoids contacting the rebar and the utility lines in order to mitigate potential damage to workers and property. Uncertain knowledge of the locations of rebar and utility lines makes drilling into reinforced concrete decks risky for worker safety and compromises the structural integrity of the deck.

4.2.1 Currently Employed Methods

One of the current methods for placing embeds into reinforced concrete bridge decks involves creating negative impressions in locations where embeds are designed to be placed. Typically, wooden dowels are used as block-outs to mark embed locations in the rebar cage. The wooden dowels are screwed into the bottom of wooden planks to hold them in place and the planks are tied to rebar chairs attached to the top layer of rebar, as shown in Figure 4.3 (Saidi et al 2011).

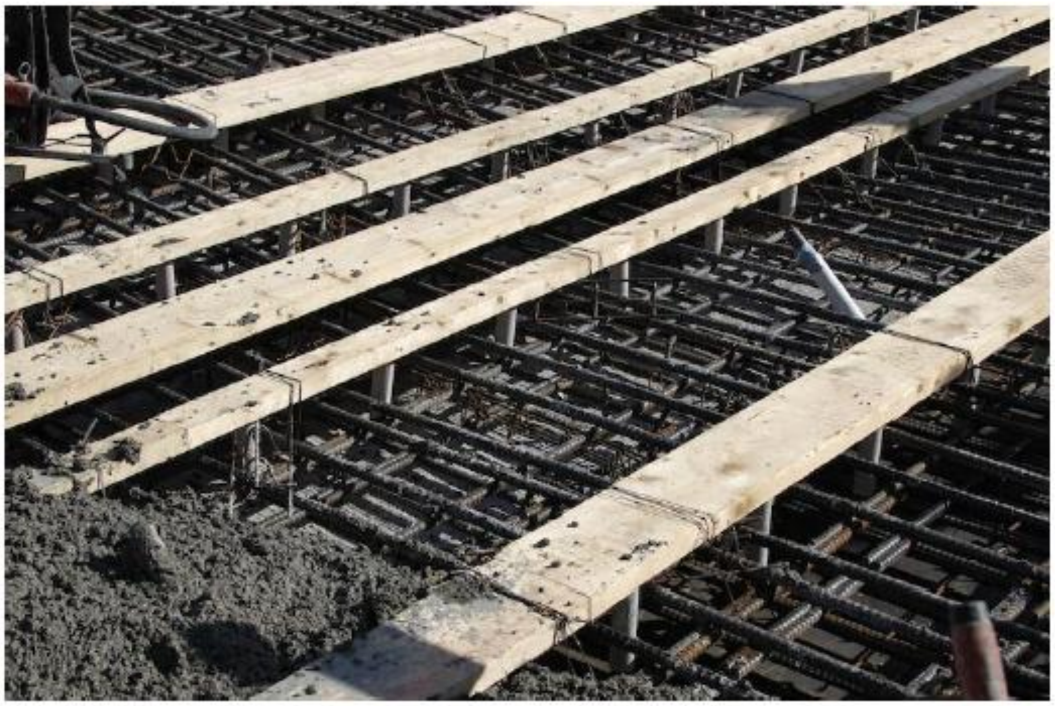


Figure 4.3: A railway bridge deck with dowels and wooden planks installed

The wooden planks are also used to create a recess along the length of the pavement and the top surface of the wooden planks is matched with the final grade of the concrete. After the concrete is placed, and after it has set for a few days, the wooden planks and dowels are removed and the holes are then covered with foam plugs to prevent debris from entering and to protect these holes from any damage, as shown in Figure 4.4 (Saidi, et al 2011).

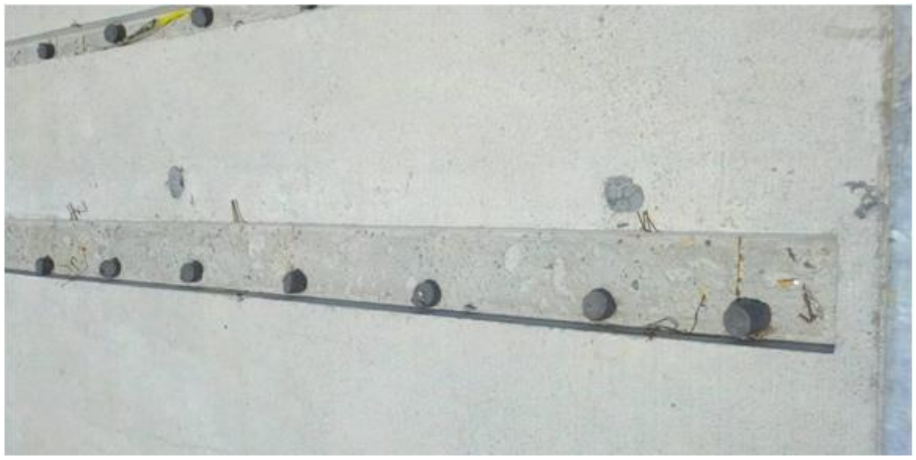


Figure 4.4: Embedment holes with block-outs removed and foam plugs inserted

4.2.2 Shortcomings of Currently Employed Methods

The process of installing the dowels and removing them is labor intensive and therefore expensive. Sometimes, the wooden planks are covered by concrete and additional time and labor are spent in locating these hidden planks. Additionally, the process also creates congestion in the rebar mats and could adversely affect the quality of the concrete by creating honeycombs and voids while restricting the access and movement of the workers placing, vibrating, and finishing the concrete (Saidi, et al 2011).

The methodology also complicates processes, such as retrofitting, rehabilitation, and drilling for additional railway tracks after the bridge deck is built, which require drilling into the bridge deck at locations not marked by the dowel block-outs. These shortcomings can be addressed by developing context-aware real-time monitoring systems that guide drilling personnel while drilling for embeds into reinforced concrete decks. The remainder of this chapter deals chiefly with the design and implementation of two such real-time context-aware systems.

4.3 Technical Approach for Controlled Drilling

4.3.1 Proposed Methodology

Based on the aforementioned framework, two potential approaches are proposed to control drilling for embeds into a reinforced concrete deck in order to prevent the drill from striking any rebar or utility lines and in order to preserve the structural integrity and utility services, respectively. The first approach is based on developing a feedback algorithm that warns the drilling personnel when a rebar or a utility line is about to be struck, as illustrated in Figure 4.5.

Before the concrete is poured, 3D imaging data is captured as a point cloud, and is then pre-processed to map the locations of the rebar and the zones safe for drilling. The position and orientation of the drill bit tip are monitored continuously in real-time, and the context-aware monitoring system determines whether the drill bit tip is about to strike the rebar or utility lines by using collision detection algorithms. If the collision detection algorithm determines that the captured information could lead to the drilling personnel striking concealed infrastructure, a warning is issued.

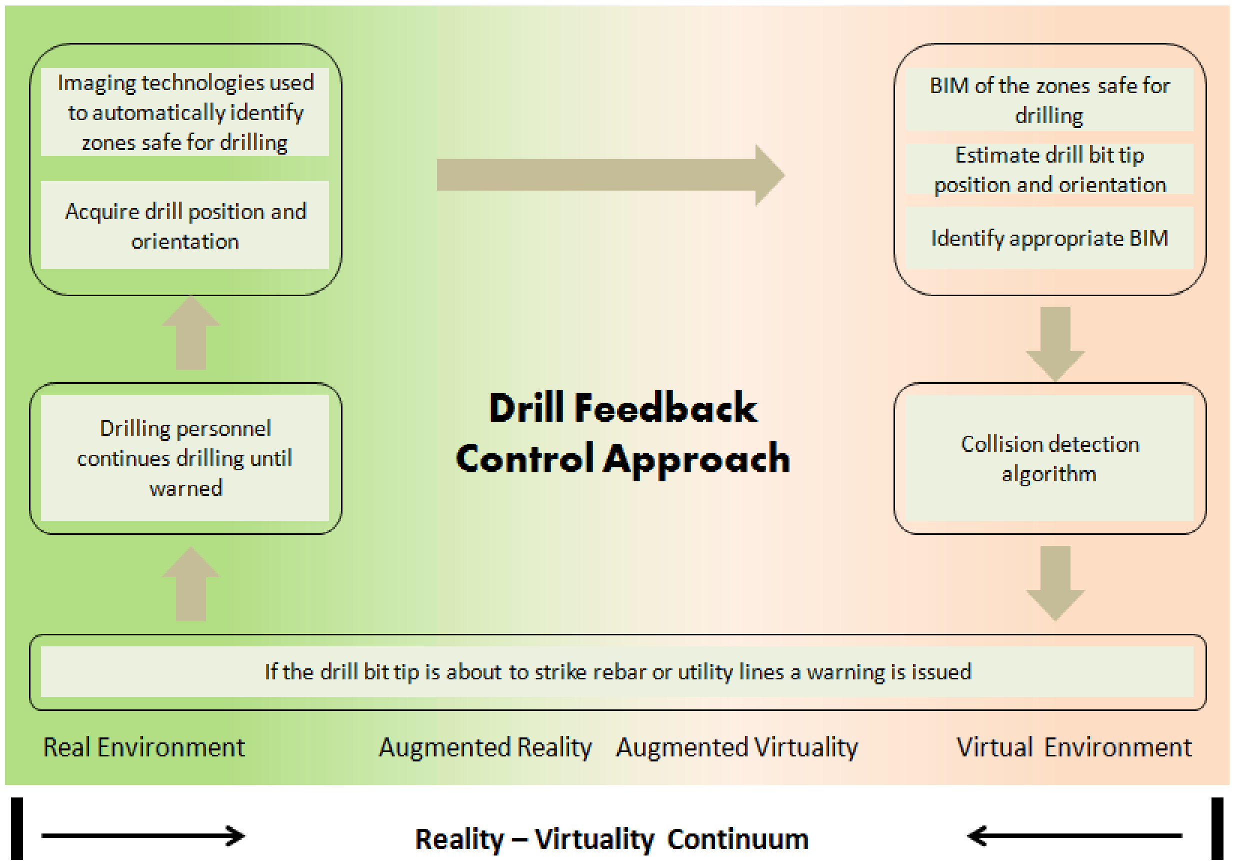

Imaging technologies used to automatically identify zones safe for drilling
Acquire drill position and orientation
BIM of the zones safe for drilling
Estimate drill bit tip position and orientation
Identify appropriate BIM
Drilling personnel continues drilling until warned
Drill Feedback Control Approach
Collision detection algorithm
If the drill bit tip is about to strike rebar or utility lines a warning is issued
Real Environment
Augmented Reality
Augmented Virtuality
Virtual Environment
Reality – Virtuality Continuum

Figure 4.5: Overview of the drill feedback control approach

The second approach is based on providing the drilling personnel with a visual representation of the locations of the rebar and the utility lines, as summarized in Figure 4.6. The locations of the rebar and utility pipes are captured by 3D imaging technologies similar to the first approach. The position and orientation of a movable laser projector is determined in real-time and the appropriate region of interest is determined. The locations of the rebar and utility lines within the region of interest are retrieved and projected onto the concrete surface to provide the drilling personnel with visual guidelines regarding the locations of the same (Akula et al 2013). Based on these visual guidelines, the drilling personnel complete the objectives in the region of interest and adjust the laser projector so that it points to the next region of interest.

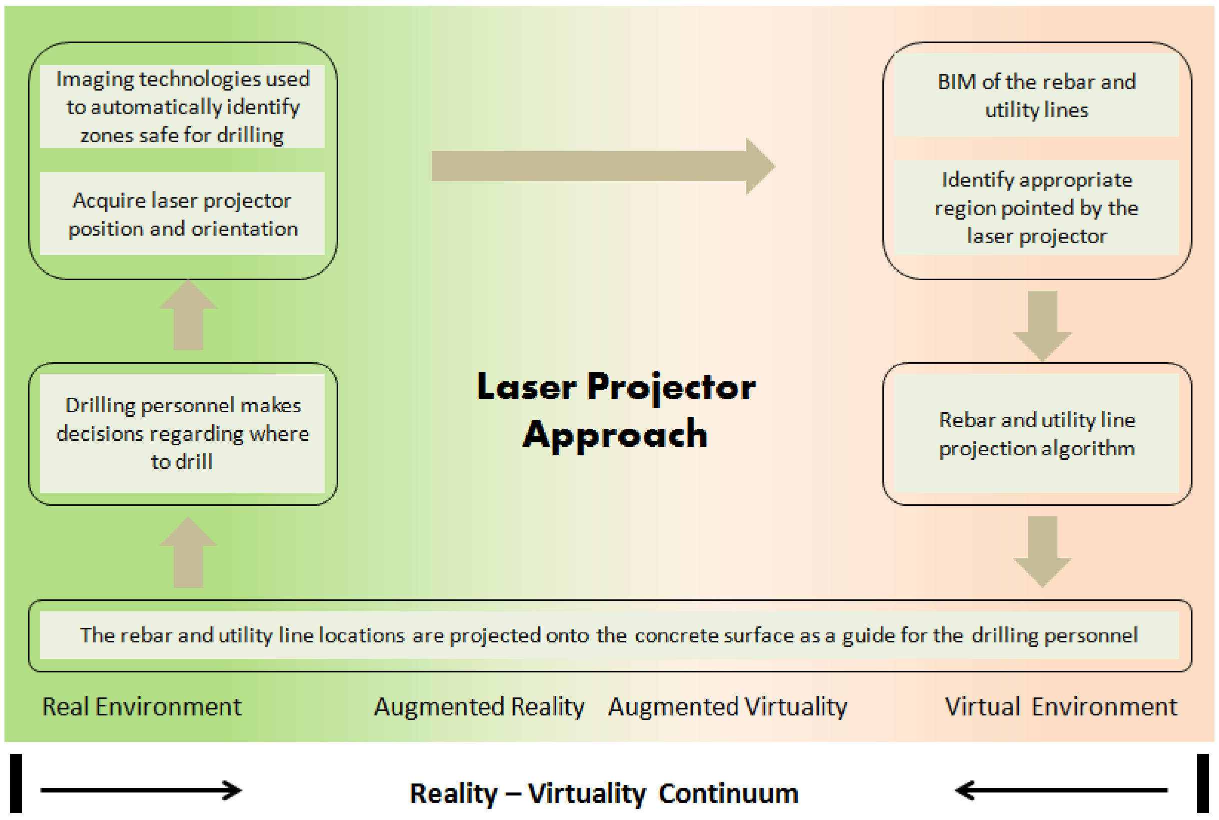

Imaging technologies used to automatically identify zones safe for drilling
Acquire laser projector position and orientation
BIM of the rebar and utility lines
Identify appropriate region pointed by the laser projector
Drilling personnel makes decisions regarding where to drill
Laser Projector Approach
Rebar and utility line projection algorithm
The rebar and utility line locations are projected onto the concrete surface as a guide for the drilling personnel
Real Environment
Augmented Reality
Augmented Virtuality
Virtual Environment
Reality – Virtuality Continuum

Figure 4.6: Overview of the laser projector approach

The approaches described above were implemented and evaluated using the Intelligent and Automated Construction Job Site (IACJS) testbed at the National Institute of Standards and Technology (NIST) using the setup described below.

4.3.2 Experimental Setup

The experimental setup for the research presented in this chapter consisted of a reconfigurable rebar cage that is a mockup of the types of rebar cages found in a railway bridge deck. The setup is shown in Figure 4.7 along with its conceptual model and pictures of similar rebar cages from an actual construction site. The reinforcement rebar cage is designed and fabricated to enable various rebar configurations. The rebar is held rigidly within the cage using clamps in order to establish a reliable ground truth. The intent of the experiment was to determine the feasibility of the concept. Therefore, the rebar cages were "clean" (i.e., without any rebar ties and other items that are typical of rebar cages in actual construction).

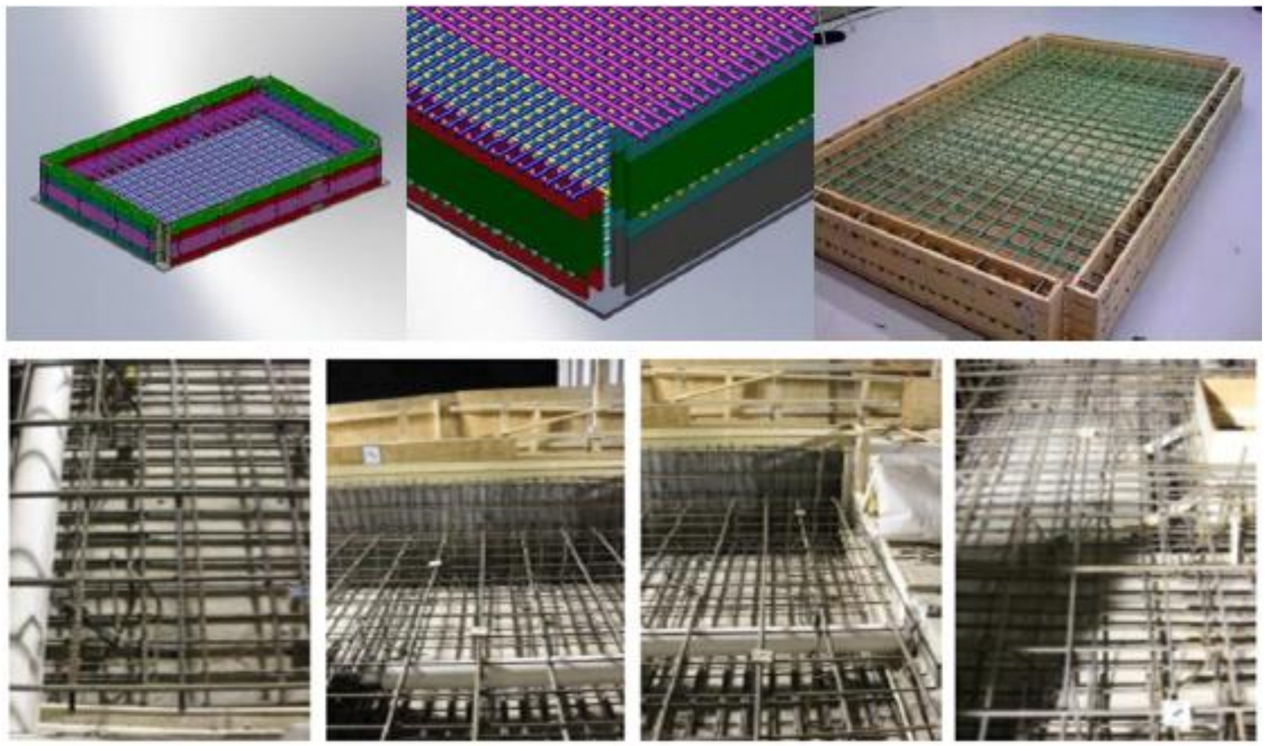


Figure 4.7: The reinforcement rebar cage 3D CAD model, as-built product, and the unordered pictures of railway deck rebar cages it represents

The as-built configuration of the rebar cage uses #6 epoxy-coated rebar (i.e., 1.9 cm diameter). Two layers of rebar are separated by approximately 30.5 cm (12 in), with each layer consisting of thirteen 3.66 m (12 ft) –long rebar laid on top of twenty-two 2.44 m (8 ft) –long rebar. The rebar are spaced at 15.2 cm (6 in) apart within each layer, and the bottom layer of rebar is also separated from the rebar cage's floor (the plywood) by 15.2 cm (6 in). The first step toward implementing and evaluating the real-time monitoring systems, proposed in the previous sections, is to acquire the digital data as point clouds. The following section describes the methods implemented to acquire the required data.

4.3.3 Acquiring Digital Data as Point Clouds

As mentioned previously, a methodology has been developed to map the locations of rebar and zones safe for drilling using 3D imaging data from laser scanning and photo reconstruction (Saidi et al 2011). A 3D imaging system, with a manufacturer-specified measurement accuracy of ±5 mm, was used to obtain the point cloud of the rebar cage configuration, as shown in Figure 4.8 (a). The point cloud was then registered to a common coordinate frame and was segmented, as shown in Figure 4.8 (b).

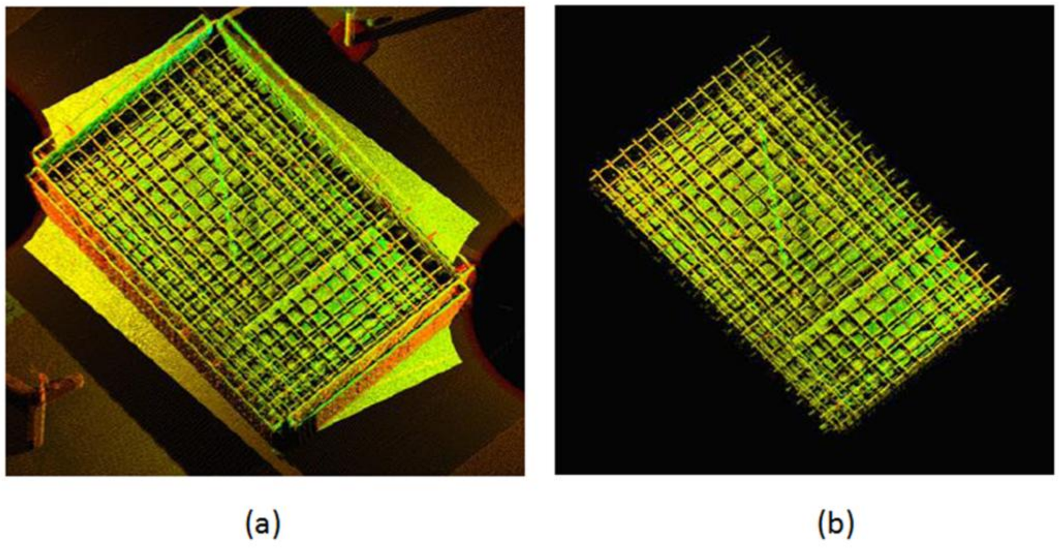


Figure 4.8: (a) The points captured by the 3D imaging scanner and (b) the segmented point cloud of the rebar cage (Akula et al 2013)

A second point cloud dataset was produced by using D4AR image-based 3D reconstruction to develop a dense point cloud (Golparvar-Fard et al 2010; Golparvar-Fard et al 2012). The point cloud was developed using 380 images of the rebar cage in which the subjects were mainly the reinforcement bars and their configurations. The spatial resolution of these images was synthetically reduced to two megapixels to test the robustness of the proposed method to low-resolution images and to minimize the computational time. The outcome of the image-based 3D reconstruction algorithm is shown in Figure 4.9 (Saidi et al 2011).

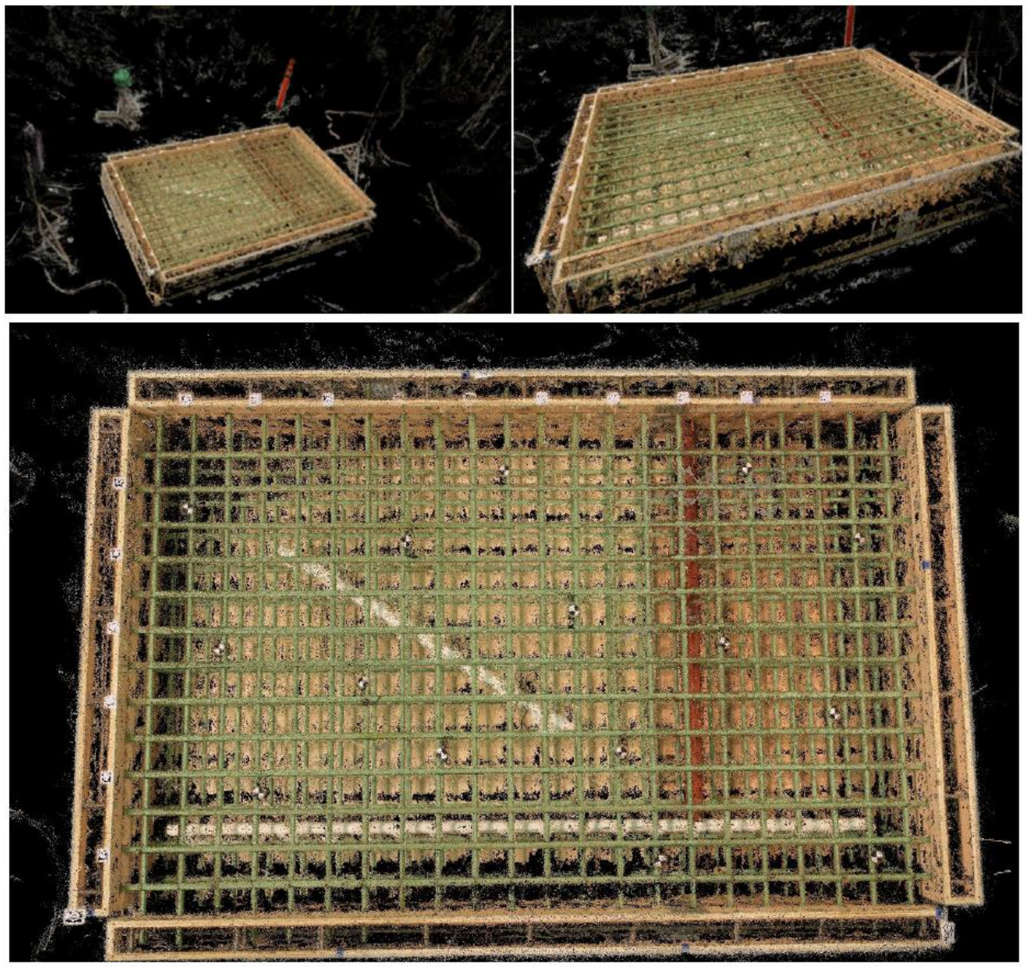


Figure 4.9: The point cloud developed using D4AR image-based 3D reconstruction (Photos courtesy Dr. Mani Golparvar-Fard)

The point cloud data to establish a ground truth was obtained using a Coherent Laser Radar (CLR) scanner instrument with a manufacturer stated uncertainty of ±100 microns and a point spacing of 3 mm (Saidi et al 2011). The point clouds were registered to a common coordinate frame. All of the point clouds were then processed through the same custom-developed rebar mapping algorithm, discussed in the following section, in order to detect the spaces between the rebar where it is safe to drill after the concrete is poured.

4.3.4 Mapping Point Cloud Data to Rebar Cage Cells

The rebar mapping algorithm involves fitting cylinders, of unfixed radii, in order to determine the intersection points of the rebar. Rebar frequently bends and deflects under its own weight and other loads. In order to account for this curvature, the rebar points are divided into shorter lengths, as highlighted in Figure 4.10, so that they can be accurately modeled as straight cylinders (Akula et al 2013).

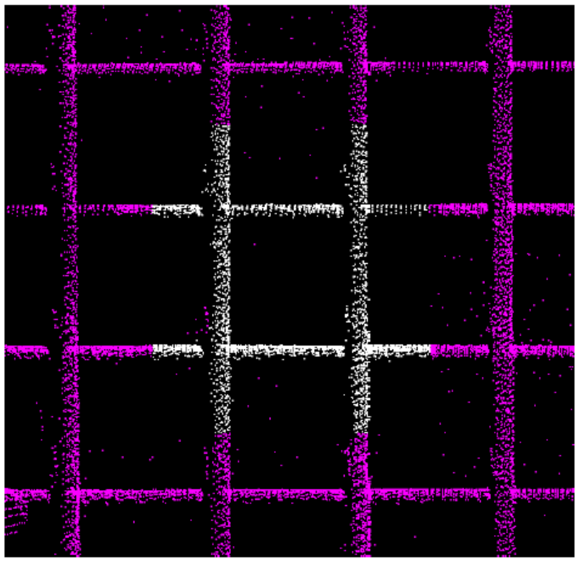


Figure 4.10: The points in white are used to fit cylinders to determine the coordinates of the rebar intersections (Akula et al 2013)

The intersections of these cylinders are determined and then projected onto a single plane parallel to the rebar layers. The rebar intersections are merged together and ordered to form a 2D grid on the aforementioned plane. Quadrilateral cells are then created on the plane with offsets equal to the radii of the cylinders modeled to fit the corresponding rebar, as shown in Figure 4.11 (Akula et al 2013).

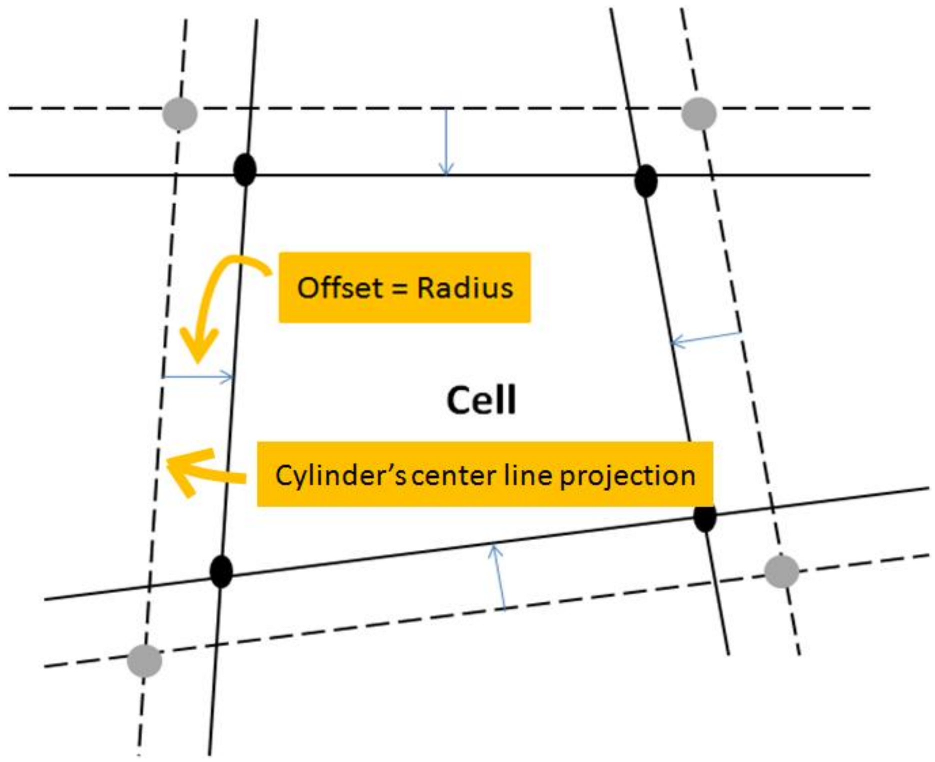


Figure 4.11: Extracting the quadrilateral cells in the rebar mapping algorithm

The algorithm then checks each cell for its safe drilling depth (i.e., the depth from the top of the concrete surface up to which drilling can be done without hitting any rebar or utility lines). The algorithm breaks the space corresponding to each rebar cell into bins and determines the number of points in each bin. If the number of points in a bin exceeds a threshold limit, then the bin is considered unsafe for drilling. Figure 4.12 shows the results of the rebar mapping and cell status prediction algorithms for a particular bin level. The number of consecutive bins deemed safe for drilling from the top of the deck determines the safe drilling depth of the cell. The rebar mapping algorithm exports a list of cells, each cell identified by the four points that make up the quadrilateral and a safe drilling depth (Akula et al 2013).

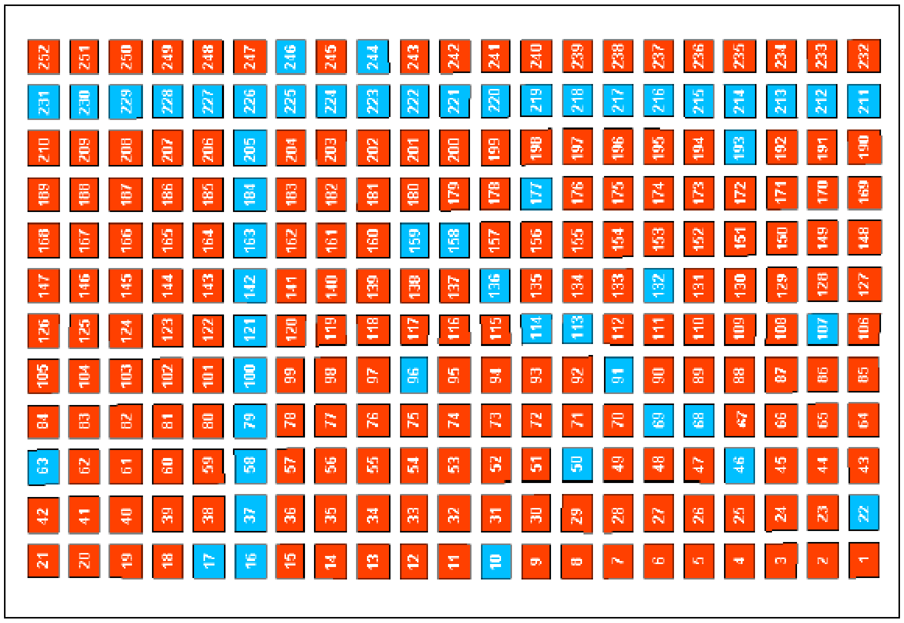


Figure 4.12: Predicted cell status for a particular bin level (Red = Safe, Blue = Unsafe) (Akula et al 2013)

Establishing the ground truth data was a manual process and was performed for each cell individually. The process involved fitting cylinders, where the radii of the cylinders are not fixed, to the point cloud of the rebar. For a given cell, only the points around that given cell are used to fit the cylinders. The cells were then classified as either safe or unsafe for drilling by visually identifying cells in which it was safe to drill versus cells in which it was not.

The results of the rebar mapping algorithm are exported as Industry Foundation Classes (IFC) files and are visualized as a BIM. IFC is an open specification object-based file format with a data model developed by buildingSMART. IFC is used to facilitate data exchange in the architecture, engineering, and construction industry, and is a commonly used format for BIM (buildingSMART 2012). The IFC representation of the rebar and bridge models is used, rather than commercial BIM software, to simplify the visualization of the models and associated safe drilling zones. The IFC files act as a proxy for BIM and could be imported into BIM software if desired. Also, the aforementioned IFC files are used as pre-processed BIM input files for the feedback, and the projection algorithms used for the controlled drilling and laser projector–based approaches.

4.3.5 Indoor Global Positioning System

The drilling feedback control and the laser projection methods, as demonstrated in the testbed, determine, in real time, whether it is safe or unsafe for a drill to continue drilling at a particular location in the concrete deck using an Indoor (or Infrared) Global Positioning System (iGPS) to track the position and orientation of the drill bit tip and the laser projector, respectively. The iGPS, shown in Figure 4.13, consists of a series of tripod-mounted transmitters that emit laser signals that are picked up by the photodiodes inside the receiver module.

Using signals from multiple transmitters, the receiver can calculate its position relative to the base coordinate system defined by the transmitters. The receiver positions can be estimated as long as they have line-of-sight to two or more transmitters; however its accuracy is improved when more than two transmitters are used for the calculation. The base coordinate system of the iGPS is registered to the common coordinate system to which the point cloud data sets were previously registered. Using an iGPS vector bar (a pair of iGPS receivers fixed relative to each other by a calibrated distance), the orientation of the vector from one receiver to the other and the 3D coordinates of any point along the vector can be computed. The frequency of updates of the iGPS measurements is 40 Hz, the uncertainty of position of each iGPS receiver is ± 0.250 mm, and the maximum allowable range between a receiver/transmitter pair is 40 m.

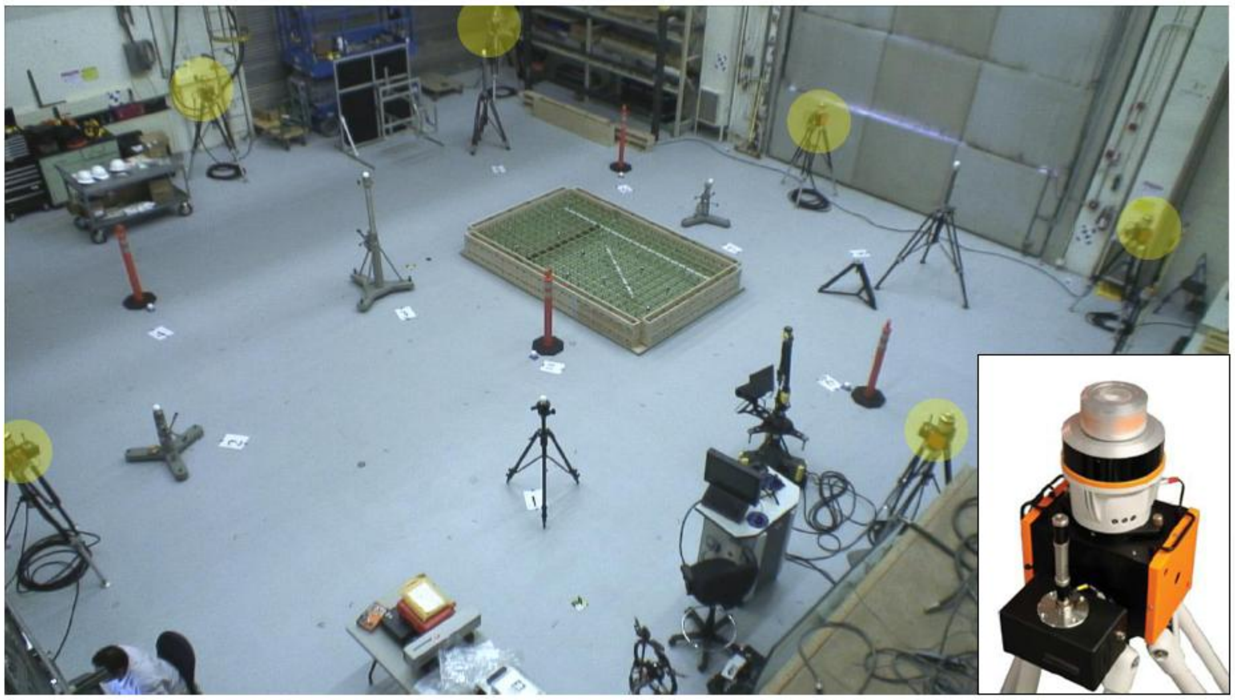


Figure 4.13: Indoor GPS with transmitters, shown in the inset image, highlighted with yellow inside the IACJS testbed at NIST

4.3.6 Drill Feedback Control Approach

The drill was fitted with an iGPS vector, as shown in Figure 4.14, mounted such that it was oriented along the longitudinal axis of the drill bit and was connected to a shoulder/waist strap, which holds the computer that performs the position calculations and communicates wirelessly with a central server.

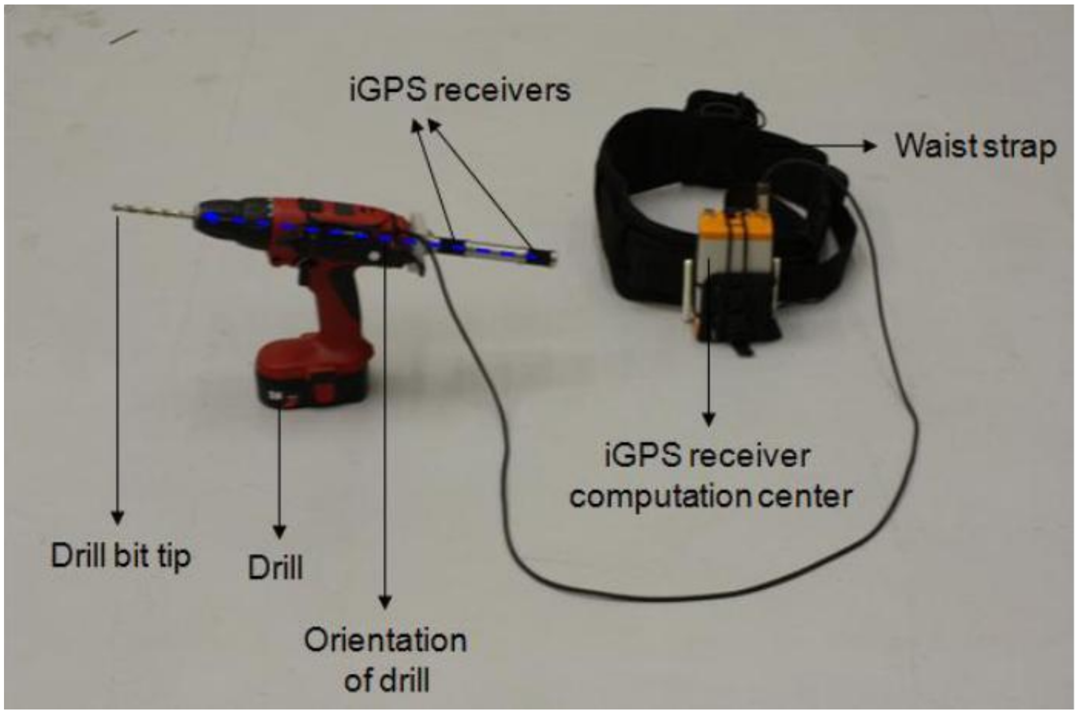


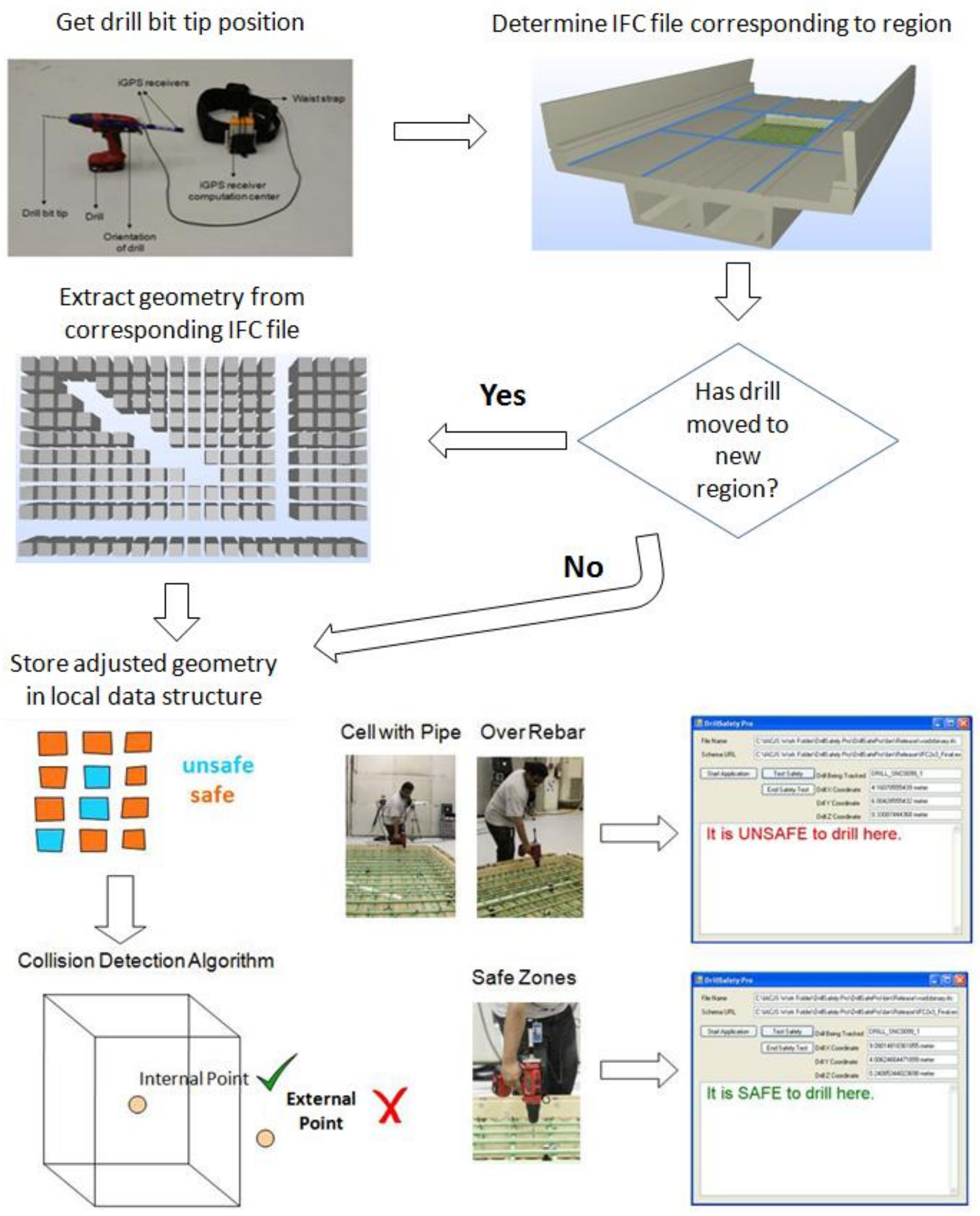
Figure 4.14: Modified drill set up with the iGPS receivers attached to the drill

The iGPS vector bar's position relative to the drill bit's tip is calibrated and preprogrammed into the iGPS server, which calculates the orientation of the drill bit and the position of its tip. The iGPS server has the ability to handle scenarios with multiple drills in order to simulate a drilling crew working simultaneously on the bridge deck. The context-aware computing application uses information regarding the position and orientation of the drill bit tip from the iGPS server and the information regarding the zones safe for drilling from the IFC data file with information regarding safe void zones as determined by the aforementioned rebar mapping algorithm.

The overall schema of the feedback application is shown in Figure 4.15. The entire reinforced bridge deck is divided into smaller regions such that each region has a corresponding BIM that stores data regarding the zones safe for drilling in that region. The application determines the drill bit position and the corresponding region, and the BIM. If the drill bit moves into a new region, corresponding to a different BIM, the BIM geometry is extracted and cell data is stored in a local data structure. These data are used by the application until the drill bit tip moves into a new region.

The application interprets the BIM data and extracts the geometry of the safe zones and stores it in a local data structure. The application interprets each safe zone as a quadrilateral prism with eight corner points. The application allows for the addition of safety tolerances to the safe zone geometry that can be defined depending on how much in advance the drilling personnel wants to be warned. The application then performs containment testing between the monitored drill bit's tip and all the safe zones to determine whether the drill bit's tip is within any of the safe void zones. This is done by first geometrically defining the perpendiculars to each face of the quadrilateral prism that face toward the inside of the cell as positive. The containment is checked for by testing whether or not the drill bit tip position is to the same side of all six planes of the quadrilateral prism.

When the drill is in a safe position (i.e., the position of the drill bit's tip is inside a safe zone), the application displays a message that it is safe to drill in that location. However, when the drill bit tip position is outside all safe zones (for example when the drill bit's tip is over a rebar or in a zone with utility pipes), the application displays a message that it is unsafe to continue drilling in that location.



It is UNSAFE to drill here.

It is SAFE to drill here.

Figure 4.15: The overall schema of the real-time drilling feedback application that determines whether is it safe or unsafe to continue drilling at a location

The application could be embedded into the drill setup to warn the drilling personnel using an audio-visual alarm. The collision detection algorithm that checks whether the drill bit tip position is in any of the safe zones within the zones (defined as such by the BIM) was found to have an update rate of over 100 Hz, making the algorithm real-time as its frequency was found to be greater than that of the input iGPS measurements.

4.3.7 Laser Projector Guidance Approach

Laser projectors can be used to visualize the locations of the rebar underneath the concrete. The as-built BIM of the rebar cage, registered with the iGPS coordinate system, was used to produce rebar patterns that help guide the drilling personnel. The position and orientation of the laser projector, shown in Figure 4.16 (a) (Saidi et al 2011), was tracked by the iGPS system. The projector can be moved and pointed at the desired locations to visualize the arrangement of the rebar underneath the concrete (Akula et al 2013).

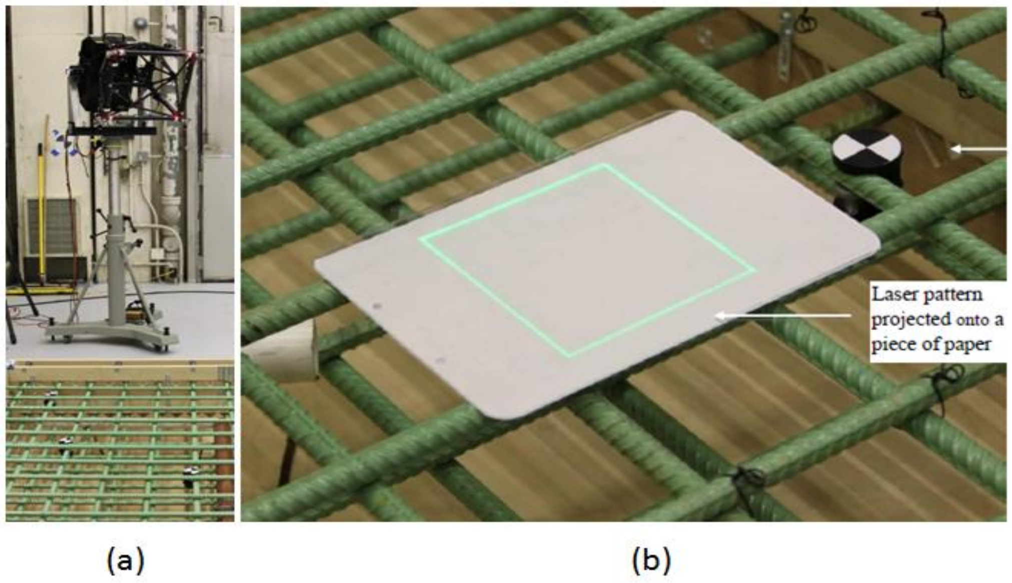


Figure 4.16: (a) The laser projector used to visualize results on the actual rebar and (b) a laser pattern projected onto a piece of paper lying on top of the rebar (Akula et al 2013)

The proposed technology was implemented and validated in the IACJS testbed by projecting patterns onto the rebar itself or onto a piece of paper that is lying flat directly on the rebar cage, as shown in Figure 4.16 (b). The projected pattern in Figure 4.16 (b) is a square corresponding to the square formed by the centerlines of the four rebar lengths directly underneath. This technique can be used as an alternate and/or complementary technology to help guide drilling into a reinforced concrete deck (Akula et al 2013).

4.4 Validation of the Methodology

The results of the rebar mapping algorithm and the cell safety depth prediction algorithm were evaluated and are presented in this section. The BIMs that were developed by the rebar mapping algorithm obtained from the laser scanning point cloud and the 3D image reconstruction point cloud were compared to the ground truth through visualization. This was done by placing them inside the CAD model of the rebar cage as part of a railway bridge deck. The first step toward achieving the visualization for comparison is the development of a bridge model and its registration to the common coordinate frame.

4.4.1 Bridge Model

A CAD model of a bridge was created to help visualize the rebar cage within the bridge structure. The cross-section of the bridge is shown in Figure 4.17 (Saidi et al 2011). The relevant bridge as-built drawings were imported into a CAD modeling software, and the cross-section of the bridge was traced to form a closed poly-line. This poly-line was extended along a curve in order to form an extruded model of the bridge. A region of space equivalent to the usable volume of the reconfigurable rebar cage was then subtracted from the bridge deck in order to model an opening in the bridge with visible rebar. The bridge deck model and the rebar cage model were grouped together and registered to a common coordinate frame and were exported as an IFC file that was used to visualize the embedded rebar cage model as shown in Figure 4.18.

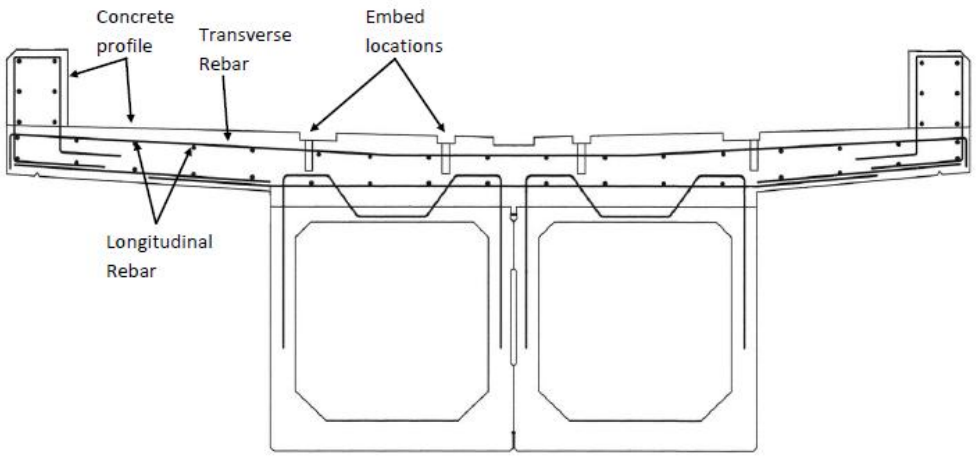


Figure 4.17: A cross section of the railway bridge deck

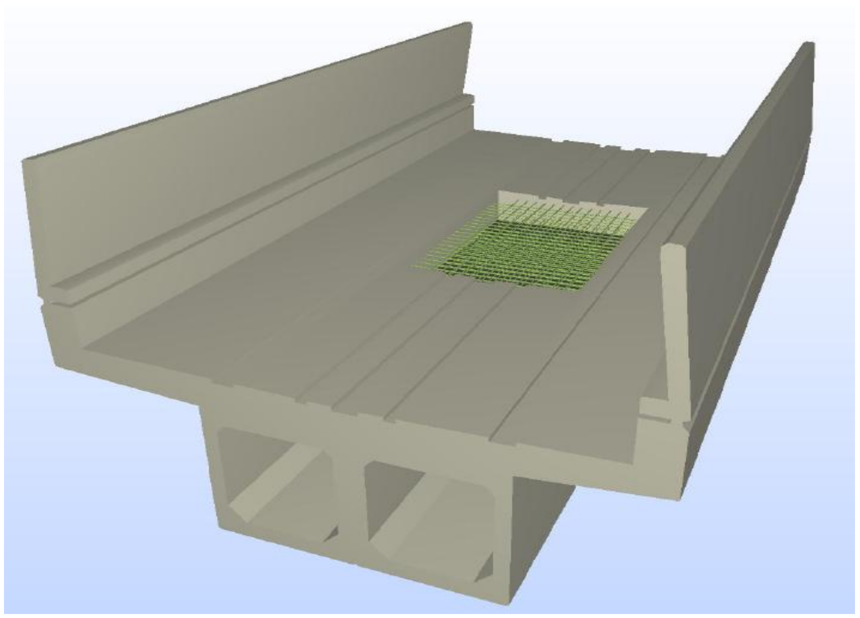


Figure 4.18: The BIM of the bridge deck with the embedded rebar cage within IFC viewer

4.4.2 Evaluation of the Rebar Mapping Algorithm

Safe and unsafe drilling zones that are computed from the rebar intersection extraction algorithm can be visualized for the ground truth, laser scan, and image-based 3D reconstruction point cloud data. The safe drilling zones are displayed along with the ideal as-designed rebar cage model. Figure 4.19 shows the perspective and overhead views of the safe drilling zones from the ground truth point cloud data (Akula et al 2013).

In the figures pertaining to this chapter, in order to make visualization more intuitive, the volumes in the BIM are only generated where it is permissible to drill through the entire depth of the rebar cage. Zones where it might be permissible to drill partially through the depth of the rebar cage are flagged as unsafe drilling zones and no volume is generated. Drilling zones are not computed along three sides of the rebar cage (left, right, and top sides in Figure 4.19 (b)) due to incompleteness of the ground truth data (Akula et al 2013).

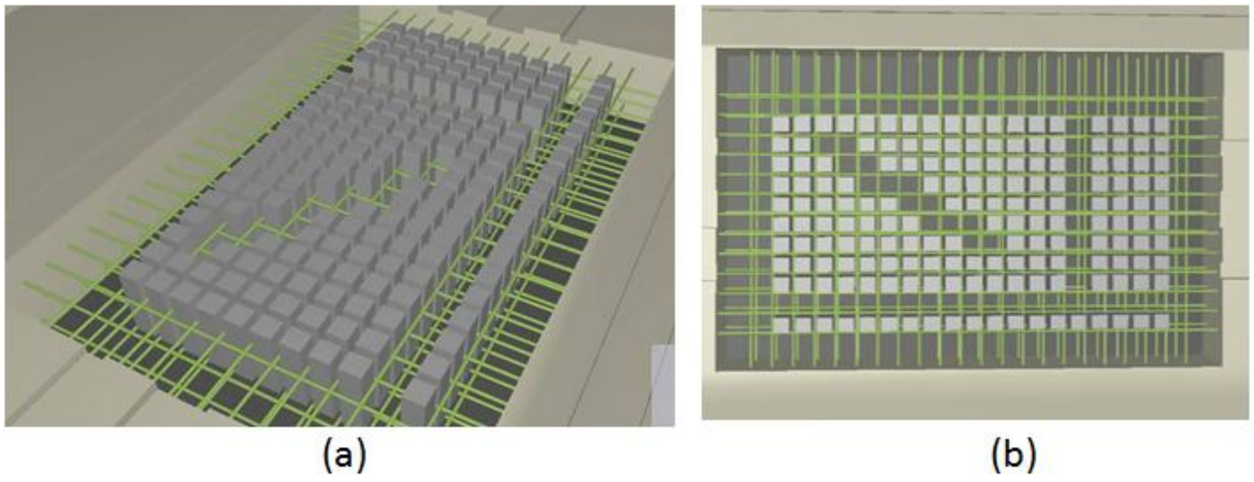


Figure 4.19: The (a) perspective and (b) overhead views of safe drilling zones from ground truth data as visualized through the IFC viewer (Akula et al 2013)

In general, the rebar cage appears between the safe drilling zones, however, due to variations of the rebar placement in the as-built rebar cage, sometimes there are clashes between certain safe drilling zones and the as-built rebar cage. Figure 4.20 shows the close-up view of the safe drilling zones obtained from ground truth data (Akula et al 2013).

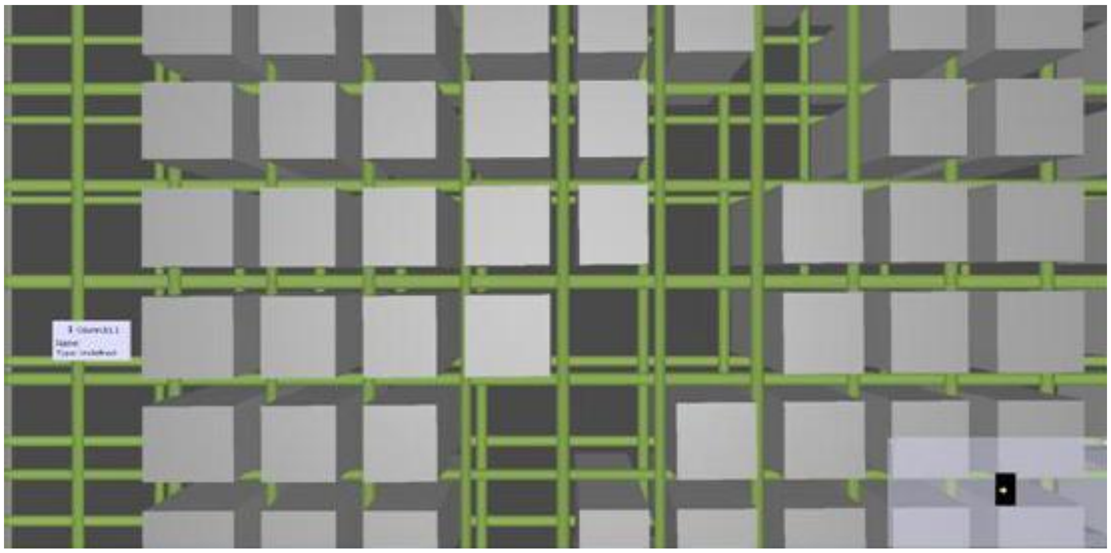


Figure 4.20: Close-up view of safe zones computed from ground truth data (Akula et al 2013)

Figures 4.21 (a) and 4.21 (b) show the safe drilling zones computed from laser scanning and image-based 3D reconstruction data, respectively. Comparing the pattern of safe drilling zones from the laser scan data with the ground truth data shows some significant differences. There are differences in safe drilling zones around the position of the diagonal conduit. The other differences in the safe drilling zones are due to rebar targets, placed in the physical rebar cage, which in some cases produce unsafe drilling zones. Comparing the safe drilling zones as predicted by the image-based 3D reconstruction data with the zones from the laser scanning data reveals a significant variation in the pattern of safe drilling zones due to missing data in the image based 3D reconstruction point cloud data set (Akula et al 2013).

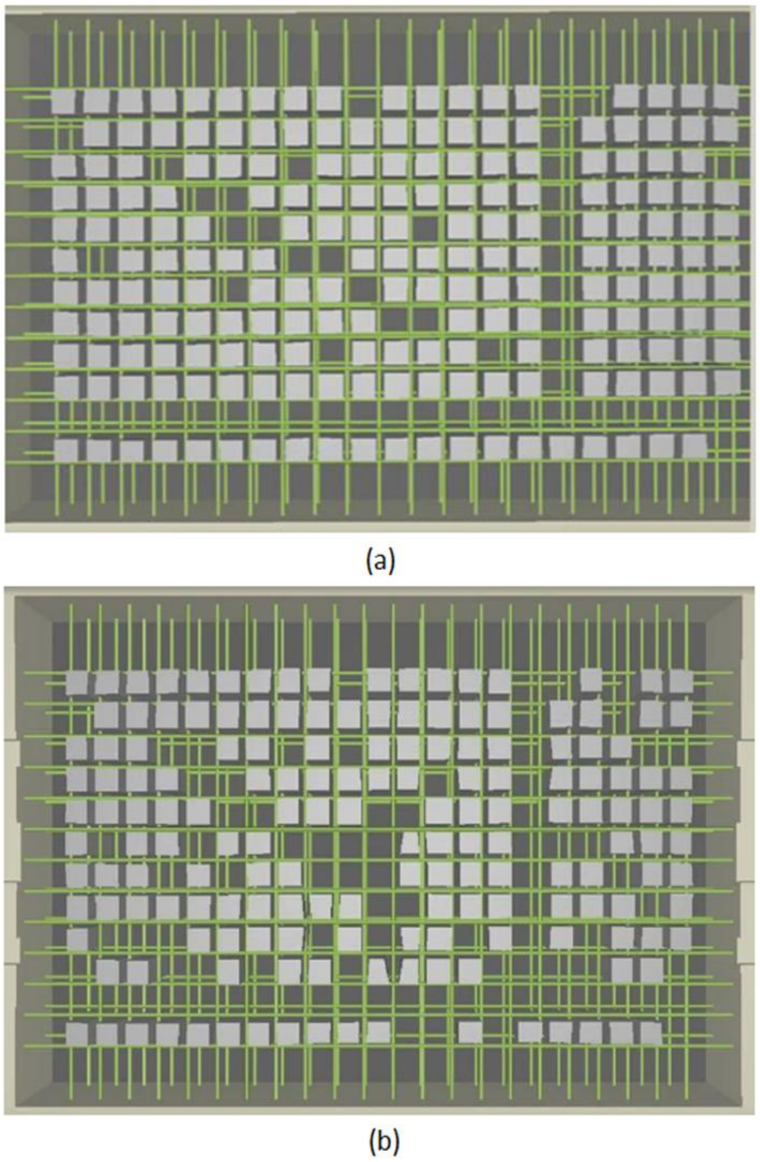


Figure 4.21: The overhead views of the safe drilling zones computed from (a) laser scan and (b) 3D image reconstruction point cloud data (Akula et al 2013)

Figure 4.22 shows the safe drilling zones computed from the ground truth data as voids in the bridge deck model (i.e., the reverse of showing the safe drilling zones as volumes in the previous figures) (Saidi et al 2011). The unsafe drilling zones are shown in blue. The portions of the reinforcement bars that are visible in the top row of as-built safe drilling zones indicate the differences between the ideal as-designed rebar cage and the position and orientation of the reinforcing bars in the as-built physical model of the rebar cage. The BIM visualized in Figure 4.22 can potentially be used as an alternative input file for the laser projection approach.

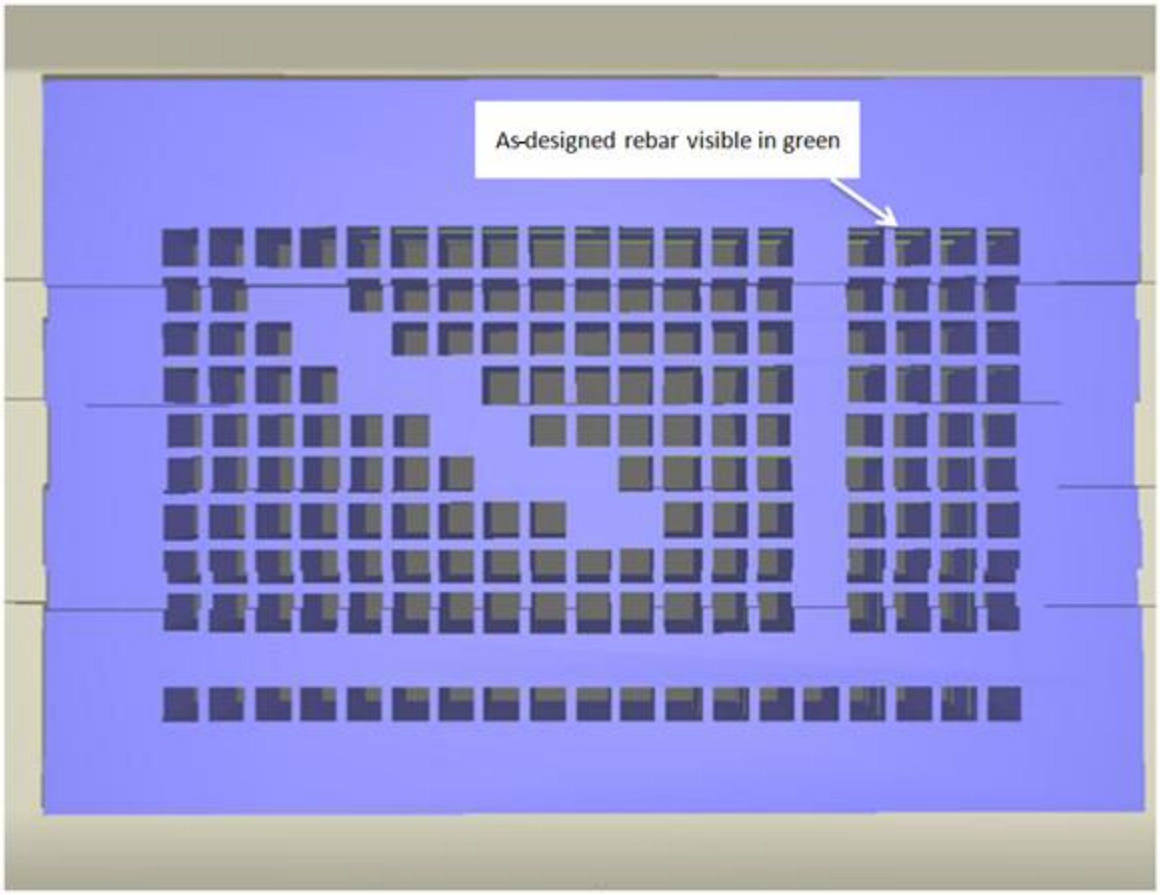


Figure 4.22: Overhead view of safe drilling zones as voids computed from ground truth data (Akula et al 2013)

Figures 4.23(a) and 4.23 (b) show the overhead view and the close-up view of the safe zone volumes for both the laser scanning (yellow) and ground truth (red) data viewed together. Figure 4.23 (b) shows the slight difference in the safe drilling zone geometry computed from each dataset. The clashing between the green reinforcing bars with the safe drilling zones, as shown in

Figure 4.23 (c), shows the differences between the as-designed and as-built reinforcing bar alignment (Akula et al 2013).

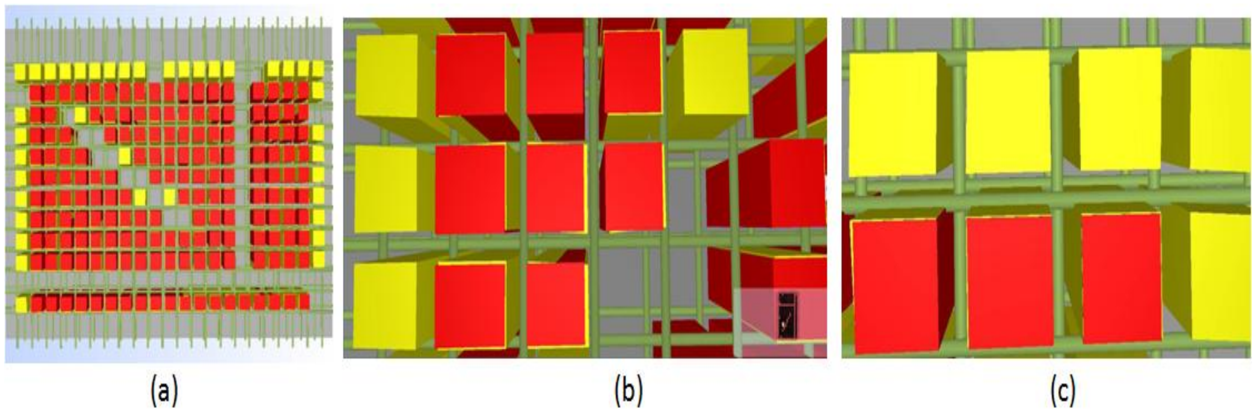


Figure 4.23: The (a) overhead and (b) close-up view of safe zone volumes for data from laser scanning (red) and ground truth (yellow) and (c) a close-up view showing overlap with as-designed rebar (Saidi et al 2011)

The visualization techniques presented above can be used to compare and evaluate the results of the rebar mapping algorithm for point clouds obtained from different technologies. The differences in the results of the rebar mapping algorithm for the laser scanning data and the ground truth data are negligible, especially when these differences are compared to the tolerances that will be applied a priori to warn the drilling personnel.

4.4.3 Results of the Cell Safe Drilling Depth Prediction

As mentioned previously, each cell is divided into bins of a fixed height and the cell's safe drilling depth is estimated based on the algorithm that predicts whether a particular bin is considered safe or unsafe based on the number of points that fall within the particular bin. If every bin in a cell is predicted to be safe, the cell is considered safe for drilling. Otherwise, the cell is considered to be unsafe for drilling. As with any binary (i.e., true/false) classifier, the possible results of the cell's safe/unsafe classifier algorithm are:

- Correct – A safe cell is identified as safe and an unsafe cell is identified as unsafe.
- False positive – An unsafe cell is identified as safe.
- False negative – A safe cell is identified as unsafe.

The results of the classifier algorithm implemented in this research for laser scanning and 3D image reconstruction data sets are summarized in Table 4.2 (Saidi et al 2011).

	Laser Scanning	3D Image Reconstruction
% of False Positives	1.2 %	0 %
% of False Negatives	0.4 %	30.5 %
% of Total False Predictions	1.6 %	30.5 %
% of Correct Predictions	98.4 %	69.5 %

Table 4.2: The results of the cell bin status classifier algorithm (Saidi et al 2011)

4.4.4 Drill Feedback Control and Laser Projector Approach

The accuracies of the drill feedback control and the laser projector-based approach depend on the accuracies of the rebar mapping algorithm and the cell bin status classifier algorithm. Moreover, they are also dependent on the accuracies of the tracking system and the safety factors included into the algorithms to warn the drilling personnel when the drill bit tip is closer to the rebar or utility lines than what is deemed acceptable. Additional safety factors—such as any tolerances used to account for the bending of the rebar due to the poured concrete and for shifting of the rebar cage—will also affect the performance of the real-time monitoring systems.

4.4.5 Limitations of the Testbed Results

The limitations of the testbed results can be classified into five main categories: data homogeneity, registration, equipment uncertainty, rebar mapping algorithm, and simulated environment limitations. These limitations are discussed below.

Data homogeneity: The point density for the various datasets varied greatly due to the different technologies used in capturing them. Also, within the dataset that is captured using a laser scanner, the point density is not spatially uniform since objects closer to the scanner generally include more points than those farther away (Akula et al 2013). In addition, the D4AR pictures were collected manually, and therefore the percent overlap between the pictures and the amount of rebar detail captured is not uniform. This leads to noise in the 3D image-based reconstruction method used to develop the point cloud (Golpavar-Fard et al 2010).

Registration: The ground truth, laser scanner, and 3D image reconstruction point clouds were registered to a common IACJS testbed coordinate frame in order to compare them with each other. The registration of the point cloud resulting from the 3D image reconstruction method to the testbed coordinate frame involved manually selecting the centers of the targets that were placed on the rebar, which resulted in unknown registration errors. The registration of the laser scanner data was more automated and used registration targets placed around the testbed.

However, the registration algorithm relies on an initial guess and the manual weighting of the targets, which also results in variations in the registration errors each time (Saidi et al 2011).

Equipment uncertainty: The uncertainty in the 3D position measurements from the iGPS and the ground truth laser scanner are ± 250 μm and ± 100 μm, respectively. Although the manufacturer specified a maximum point uncertainty of ± 5 mm for the other laser scanner used, the uncertainties in the data obtained from that laser scanner are unknown due to other sources of error such as angle-of-incidence and optical properties of the materials in the rebar cage. The uncertainties of the data obtained using the camera hardware and the D4AR software combination are also unknown. Furthermore, it is also unknown how the registration process affects the uncertainty of the data. Current methods of quantifying these errors (i.e., how registration errors propagate into all of the measured data points) are inadequate because they assume that the error in each data point in the registered point cloud is the same as the mean error around the registration targets.

Rebar mapping algorithm: The algorithm assumes that once the concrete is poured on top of the rebar, the resulting concrete surface will be a flat plane parallel to the floor of the rebar cage and that the process of pouring the concrete will not affect the locations of the rebar and utility lines. The algorithm would have to be extended in order to account for more complex finished concrete surfaces, and appropriate safety factors and tolerances must be introduced to account for changes in the rebar and utility line locations due to bending and other side effects of the concrete pouring process (Akula et al 2013; Saidi et al 2011).

Simulated environment: The results from the experiments conducted in the testbed described in the chapter are also limited by the fact that all the experiments were conducted in a clean and controlled laboratory environment using a mockup of a rebar cage. However, the field conditions are often more congested and cluttered and the site is much larger than the laboratory setting leading to data occlusions due to objects (e.g., people, equipment, tools, general clutter) being in the way and to the limited accessibility to locations for setting up the instruments or taking photos. Rebar cages also often contain objects or items that introduce noise to the data. Items such as rebar ties, rebar chairs, formwork, trash, and tools are often found in or on rebar cages under field conditions. Rebar hooks, bends, and splices will also add complexity. The ability to handle the added noise and complexity would require the development of a better point cloud filtering algorithm and a more sophisticated classifier algorithm.

Tracking system: Implementing iGPS is impractical for an actual construction worksite and certainly less so for a bridge deck. Both iGPS and laser projectors would suffer under direct sunlight due to the fact that the iGPS uses infrared signals and the projector's laser intensity may not be sufficient. Also, the dust and the less-than-ideal conditions of a construction site may render both systems unusable. The testbed implementation of real-time monitoring methods was a demonstration of the concept behind using such a framework—not a demonstration of how they could actually be used in construction conditions (Saidi et al 2011). A rugged tracking system, with the required accuracies, that works under actual field conditions and provides similar functionality to the iGPS system does not currently exist and would have to be developed. When and if better and more rugged tracking and visualization technology becomes available, these methods could be applied in the field. The cost of such a system would also have to be taken into consideration (Akula et al 2013).

4.5 Summary and Conclusions

Real-time context-aware monitoring systems have the potential to improve the efficiency of construction operations and to significantly reduce the loss of life and property in construction. This chapter presented a framework for developing real-time monitoring systems based on context-aware computing techniques that identify hazardous scenarios and help construction personnel make more informed decisions.

The author presented a motivating scenario of drilling for embeds into reinforced concrete bridge decks where monitoring systems could have a significant impact in avoiding striking concealed rebar and utility infrastructure. The author then proposed two potential methods to address the problem of monitoring the process of drilling for embeds in real-time. The two approaches—the drill feedback control approach and the laser projector-based guidance approach—were then implemented in a testbed using a rebar cage designed as a mockup of a railway bridge rebar cage.

The results showed the feasibility of both approaches to provide real-time feedback for drilling into the concrete. The shortcomings of the proposed real-time monitoring methods developed in the testbed and the challenges of implementing them in the field were discussed. The methods presented could significantly improve production for the concrete deck placement operation, avoiding the time and cost to place and remove dowels, and shorten the project duration (Saidi et al 2011).

4.6 Acknowledgments

The presented work was partially conducted at NIST using the IACJS testbed. The author gratefully acknowledges NIST's support. The author would also like to thank Dr. Kamel Saidi, Dr. Geraldine S. Cheok, Dr. Marek Franaszek, Prof. Mani Golpavar-Fard, Behshad Ghadimi, and Dr. Robert R. Lipman for their assistance in conducting some of the experiments. Any opinions, findings, conclusions, and recommendations expressed in this chapter are those of the author and do not necessarily reflect the views of the NIST or the individuals mentioned here.

4.7 References

Abderrahim, M., Garcia, E., Diez, C., and Balaguer, C. (2005). “A Mechatronics Security System for the Construction Site,” Automation in Construction, Vol. 14(4), pp. 460–466.

Akula, M., Lipman, R.R., Franaszek, M., Saidi, K.S., Cheok, G.S., and Kamat, V.R. (2013a). “Real-Time Monitoring: Augmenting Building Information Modeling with 3D Imaging Data to Control Drilling for Embeds into Reinforced Concrete Bridge Decks,” Automation in Construction (In Review).

Arditi, D., Ayrancioglu, M. and Shu, J. (2005). “Worker Safety Issues in Nighttime Highway Construction,” Engineering Construction and Architectural Management, Vol. 12(5), pp. 487–501.

buildingSMART (International Alliance for Interoperability, IAI) (2012). "International Home for openBIM," Website of buildingSMART <http://www.buildingsmart.com/> as accessed on June 21st, 2012.

Bureau of Labor Statistics (BLS) (2010). "Census of Fatal Occupational Injuries (CFOI) - Current and Revised Data," Website of the United States Bureau of Labor Statistics: Injuries, Illnesses and Fatalities < http://www.bls.gov/iif/oshcfoi1.htm#2009> as accessed on June 21st, 2012.

Burrell, J. and Gay, K. (2001). "Collectively Defining Context in a Mobile, Networked Computing Environment," Proc. of the Conference on Human Factors in Computing Systems, Association for Computing Machinery (ACM), New York, NY, pp. 231–232.

Chae, S., and Yoshida, T. (2010). "Application of RFID Technology to Prevention of Collision Accident with Heavy Equipment," Automation in Construction, Vol. 19(3), pp. 368–374.

Cheng, T., and Teizer, J. (2011). "Crane Operator Visibility of Ground Operations," 2011 Proc. of the 28th Int. Symp. on Automation and Robotics in Construction (ISARC 2011), Seoul, Korea, pp. 699–705.

Fullerton, C. E., Allread, B. S., and Teizer, J. (2009). “Pro-Active Real-Time Personnel Warning System,” Proc. of the 2009 Construction Research Congress, American Society of Civil Engineers (ASCE), Reston, VA, pp. 31–40.

Golparvar-Fard, M., Peña-Mora, F., and Savarese, S. (2010). “D4AR - 4 Dimensional Augmented Reality-Tools for Automated Remote Progress Tracking and Support of Decision-Enabling Tasks in the AEC/FM Industry,” Proc. of the 6th Int. Conference on Innovations in AEC Special Session - Transformative Machine Vision for AEC. State College, PA.

Golparvar-Fard, M., Ghadimi, B., Saidi, K. S., Cheok, G. S., Franaszek, M. and Lipman, R. R. (2012). “Image-Based 3D Mapping of Rebar Location for Automated Assessment of Safe Drilling Areas Prior to Placing Embedments in Concrete Bridge Decks,” Proc. of the 2012 Construction Research Congress, American Society of Civil Engineers (ASCE), West Lafayette, IN, pp. 960–970.

Hinze, J. W., and Teizer, J. (2011). “Visibility-Related Fatalities Related to Construction Equipment,” Safety Science, Vol. 49, pp. 709–718.

National Institute for Occupational Safety and Health (NIOSH). (2007). “Fatality Assessment and Control Evaluation (FACE) Program,” Website of the National Institute for Occupational Safety and Health < http://www.cdc.gov/niosh/face/> as accessed on June 21st, 2012.

Ruff, T. M. (2008). “Recommendations for Evaluating and Implementing Proximity Warning System on Surface Mining Equipment,” Research Report No. 2007-146, National Institute for Occupational Safety and Health (NIOSH), Atlanta.

Saidi, K., Cheok, G., Franaszek, M., Brown, C., Swerdlow, J., Lipman, R., Katz, I., Golparvar-Fard, M., Goodrum, P., Akula, M., Dadi, G., and Ghadimi, B. (2011). “Report on the Development and Use of the NIST Intelligent and Automated Construction Job Site Testbed,” National Institute of Standards and Technology (NIST), Gaithersburg, MD.

Schilit, B. N., Adams, N., and Want, R. (1994). “Context-Aware Computing Applications,” Proc. of the Workshop on Mobile Computing Systems and Applications, Institute of Electrical and Electronics Engineers (IEEE), Santa Cruz, CA, pp. 85–90.

Son, H., Kim, Ch., Kim, H., Choi, K.-N. and Jee, J.-M. (2008). “Real-Time Object Recognition and Modeling for Heavy-Equipment Operation,” Proc. of the 25th Int. Symp. on Automation and Robotics in Construction (ISARC 2008), Vilnius, pp. 232–237.

Talmaki, S., Dong, S., and Kamat, V. R. (2010). "Comprehensive Collision Avoidance System for Buried Utilities,” Proc. of the 2010 Int. Conference on Sustainable Urbanization (ICSU), Hong Kong Polytechnic University, Hong Kong, pp. 1265–1274.

Teizer, J., Allread, B.S. and Mantripragada, U. (2010). “Automating the Blind Spot Measurement of Construction Equipment,” Automation in Construction, Elsevier, Vol. 19 (4), pp. 491–501.

Wu, W., Yang, H., Chew, D. A. S., Yang, S., Gibb, A. G. F., and Li, Q. (2010). “Towards an Autonomous Real-Time Tracking System of Near-Miss Accidents on Construction Sites,” Automation in Construction, Vol. 19(2), pp. 134–141.

Zhang C., Hammad, A. and AlBahnassi H. (2008). “Collaborative Multi-Agent Systems for Construction Equipment based on Real-Time Field Data Capturing,” Next Generation Construction IT: Technology Foresight, Future Studies, Roadmapping, and Scenario Planning, Vol. 14, pp. 204–228.

Chapter 5

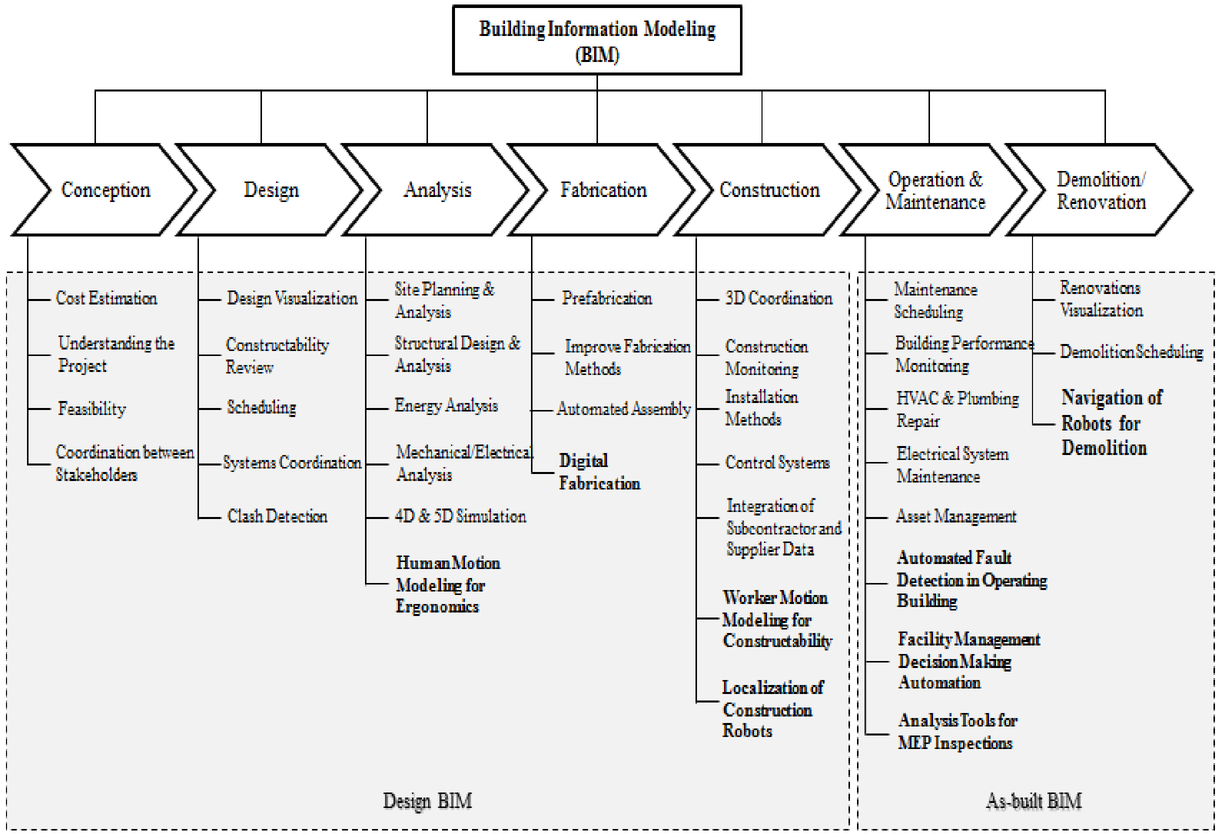
Automated Fault Detection in Operational Buildings

5.1 Facility Management in Operational Buildings

Facility Management (FM), as defined by the International Facility Management Association (IFMA), is a profession that encompasses multiple disciplines to ensure functionality of the built environment by integrating people, places, processes, and technology. FM practices contribute to 5%–10% of the gross domestic product in developed countries (Madritsch et al 2009) and the total life cycle costs of a facility could be as much as 7 times higher than the initial investment costs (Seul-Ki et al 2012). Moreover, more than 85% of the total costs of ownership are spent on FM activities (Teicholz 2004). Considering this significant cost associated with the operations and maintenance (O & M) phase of facilities, the importance of developing efficient FM practices is clearly understood in the industry and among owners. Since FM activities are information intensive and all related decision-making processes involve vast amounts of data, having real-time and efficient access to information is essential to making a facility's maintenance more feasible. However, due to the communication gap between the contractor and the owner, FM processes are usually time consuming and laborious. Upon completion of a project, the contractor hands over to the owner an enormous amount of as-built information. Then, the owner/facility manager spends a significant amount of time and money on the documents to implement the information into a facility management system. This is mainly due to the incompatibility of the information systems supporting FM practices. According to the National Institute of Standards and Technology (NIST), the capital facilities construction industry wastes $15.8 billion annually due to interoperability inefficiencies. Of that figure, $5.2 billion is attributed to the participants within the Architecture, Engineering, and Construction (AEC) industry, and the remaining $10.6 billion loss is attributed to the owner/facility manager during the O & M phase of the facility (Newton 2004).

5.2 Building Information Modeling

The National Building Information Model Standard Project Committee defines Building Information Modeling (BIM) as a digital representation of the physical and functional characteristics of a facility (BuildingSMART Alliance 2013). BIM is a value creating process that involves the generation, management, and exchange of knowledge of a facility forming a reliable basis for decision making throughout its life cycle—from the conceptual, design, and construction phases, through its operational life and subsequent closure (East and Brodt 2007). As a comprehensive digital database, BIM is gaining acceptance as the preferred tool of communication during the design and construction phases. While the use of BIM is usually limited to design and construction, owners and facility managers have realized its potential and persist on inheriting BIM models for further use. Figure 5.1 summarizes the current and potential uses of BIM in different phases of a construction project during its life cycle (Golabchi et al 2013). The bold text in Figure 5.1 indicates the potential areas and applications that researchers have identified where the proliferation of BIM can be highly beneficial (Golabchi et al 2013).



Building Information Modeling (BIM)
Conception
Design
Analysis
Fabrication
Construction
Operation & Maintenance
Demolition/ Renovation
Cost Estimation
Understanding the Project
Feasibility
Coordination between Stakeholders
Design Visualization
Constructability Review
Scheduling
Systems Coordination
Clash Detection
Site Planning & Analysis
Structural Design & Analysis
Energy Analysis
Mechanical/Electrical Analysis
4D & 5D Simulation
Human Motion Modeling for Ergonomics
Prefabrication
Improve Fabrication Methods
Automated Assembly
Digital Fabrication
3D Coordination
Construction Monitoring
Installation Methods
Control Systems
Integration of Subcontractor and Supplier Data
Worker Motion Modeling for Constructability
Localization of Construction Robots
Maintenance Scheduling
Building Performance Monitoring
HVAC & Plumbing Repair
Electrical System Maintenance
Asset Management
Automated Fault Detection in Operating Building
Facility Management Decision Making Automation
Analysis Tools for MEP Inspections
Renovations Visualization
Demolition Scheduling
Navigation of Robots for Demolition
Design BIM
As-built BIM

Figure 5.1: Potential applications of BIM during a facility's lifecycle (Golabchi et al 2013)

The use of BIM during the design and construction phases has received considerable attention in the AEC field from the research community (Azhar et al 2008; Eastman et al 2008; Goedert and Meadati 2008; Khemlani 2009), as well as the professional community (Becerik-Gerber et al 2012; Smith and Tardif 2009), and has evolved into a mature practice. The use of BIM during the design phase for maintainability, clash detection (Dunston and Williamson 1999; Eastman et al 2008; Leite et al 2009), and energy sustainability analysis (Barnes 2009; Cho et al 2010; Miller 2010; Stumpf et al 2009) has shown significant promise. BIM has been proposed as a platform for multi-disciplinary collaboration (Singh et al 2011; Tatum and Korman 2000) and project control (Hwang and Liu 2010) during construction. Several problems that occur with the current procedure for construction hand-over documents and the improvements that BIM provides have also been investigated (East and Brodt 2007). The process of capturing accurate as-built BIM data—for buildings in the construction and post-construction phases—has also received significant attention in recent years (Huber et al 2011; Liu 2012; Woo et al 2010).

The potential benefits of implementing BIM in the O & M phase are widely accepted among researchers (Motawa and Almarshad 2012), and related research indicates the industry's tendency to take advantage of the new and unique opportunities that as-built BIM offers (Becerik-Gerber et al 2012). BIM has the potential to improve communication and interoperability in FM by eliminating inefficiencies and streamlining the O & M systems for facilities (Akcamete et al 2010). The concept of Facilities Management Classes (FMC) —a collection of object class definitions for representing the information used in carrying out FM activities—has been developed to support the sharing and exchange of information among FM applications within an integrated environment (Yu et al 2000).

Opportunities for utilizing BIM for facility management have also been investigated with a focus on supporting maintenance planning (Akcamete et al 2010) and identifying useful information in a BIM model needed by facility managers (Anoop et al 2011). Wireless Sensor Networks have been integrated with BIM to monitor physical and environmental conditions during the operation stage of the building (Eastman et al 2008). However, despite the clear benefits of using BIM in FM practices, research related to utilizing BIM in FM is still in its infancy. Researchers believe that developing and introducing BIM-based applications for FM is essential for increasing the adoption of BIM in the O & M industry (Golabchi et al 2013). The author believes that the facilitation of decision making for FM purposes is one of the main areas in which the use of BIM can be highly beneficial.

5.3 Motivating Scenario

5.3.1 Heating, Ventilation, and Air Conditioning System Failure

For the purpose of this study, the author investigated the process of inspecting and repairing damaged Heating, Ventilation, and Air Conditioning (HVAC) equipment as implemented at UM Plant Operations. The UM Plant Operations office is the resource for all facility maintenance services at the University of Michigan. The office provides around-the-clock building maintenance, operation, and environmental monitoring for over 30 million sq. ft. of facilities that serve the university campuses, hospital, and health centers. One of the main tasks of the office is the maintenance and repair of the Mechanical, Electrical, and Plumbing (MEP) systems and equipment—from electric outlets or thermostat in offices, to a 1200-ton centrifugal chiller—including HVAC systems.

Upon HVAC system failure, the facility occupants report deterioration in performance to the UM Plant Operations office's ticketing database. UM Plant personnel, responsible for HVAC maintenance, reconcile the reported malfunction with the facility's paper plans, blueprints, specifications, and inventory documentation. The personnel then try to localize the potentially failed equipment and generate a plan of action based on their judgment and experience. The plan of action is used as a guide by field personnel to inspect the HVAC system for failed equipment and to make repairs if failure is detected.

5.3.2 Shortcomings of Currently Employed Methods

This process—as currently implemented across the industry—is labor intensive, time consuming, and often susceptible to unreliable or outdated information. While executing the maintenance and repair tasks, FM personnel need real-time access to vast amounts of facility information. By using BIM models instead of paper blueprints, FM personnel can reconcile real components with corresponding 3D models and guide themselves through the system to promptly execute the plan of action. Integrating BIM with FM databases also makes the repair process more efficient by providing field personnel with relevant information such as specifications and the history of the equipment's failure.

While real-time visualization of equipment information and topology helps FM personnel in streamlining their operations, the true potential of BIM is realized as an analytical tool to support decision making in FM. This chapter proposes automating the process of fault detection and the generation of the plan of action for field inspectors. Adopting automated fault detection and plan of action generation, as described in the following section, enables FM field personnel to promptly access relevant information regarding failed equipment, thus saving considerable repair time and effort.

5.4 Research Methodology

5.4.1 HVAC Distribution Network Model

The research presented in this chapter was conducted by investigating the HVAC system in a section of the G.G. Brown Building at the University of Michigan, as shown in Figure 5.2.

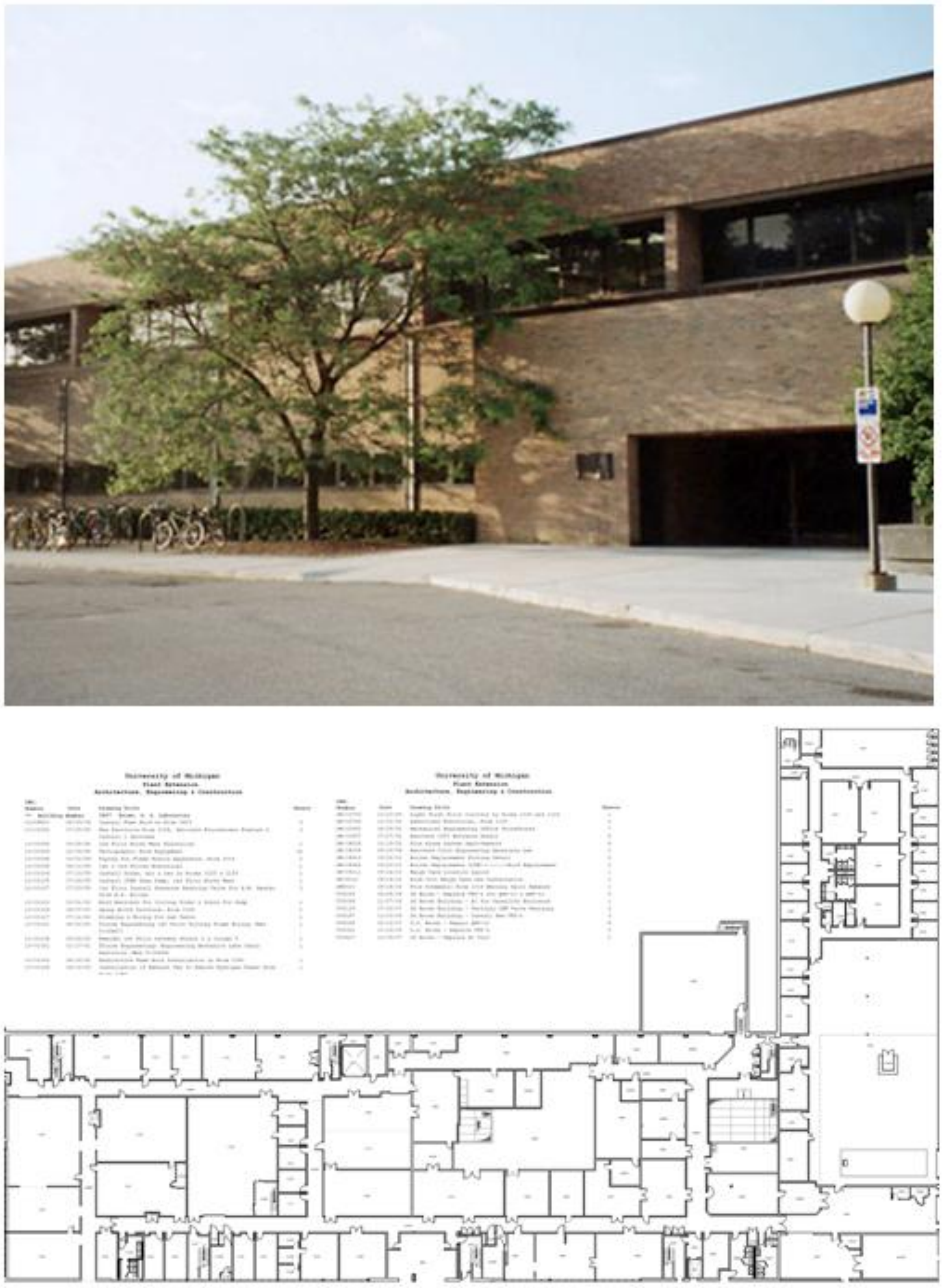
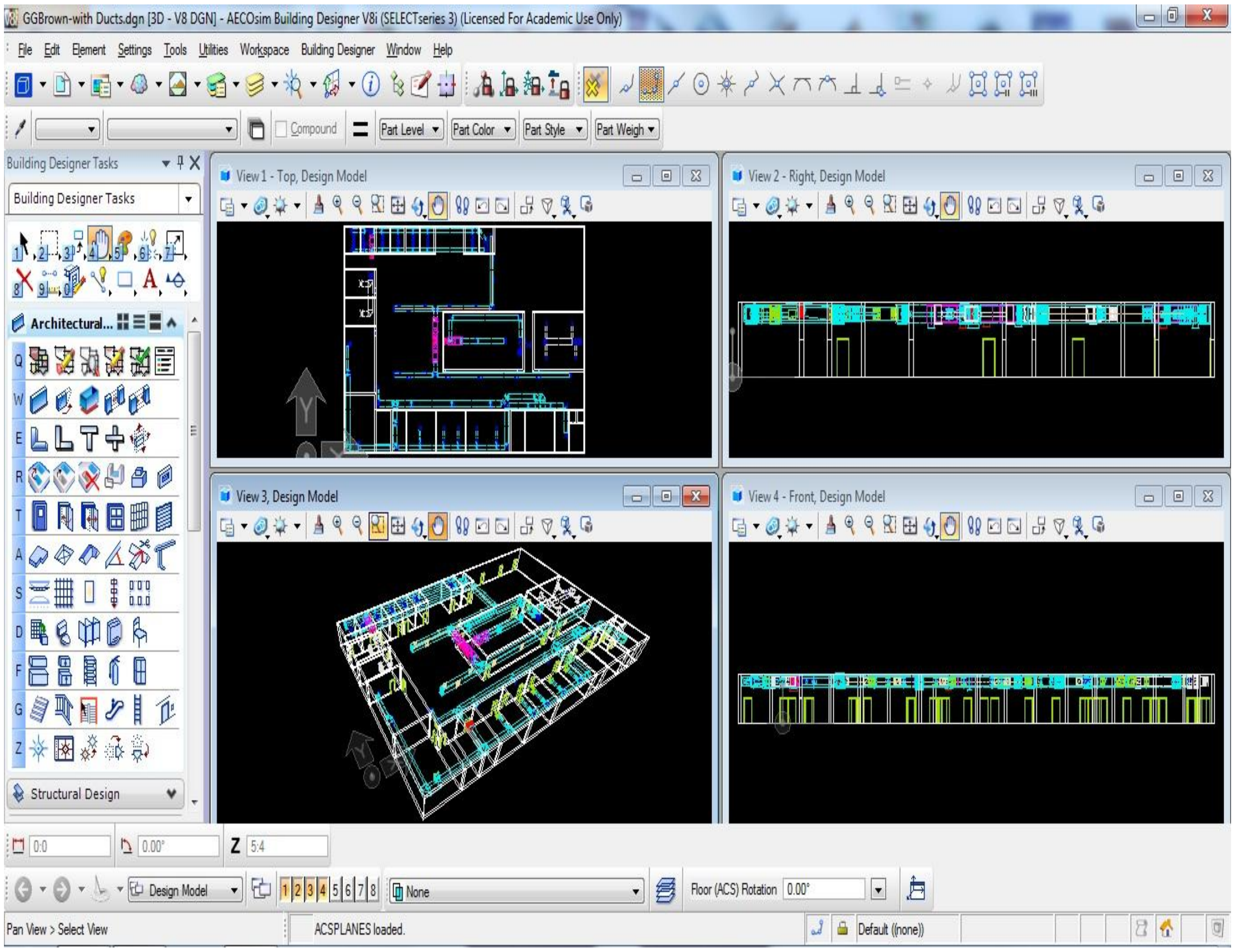

University of Michigan
Plant Extension
Architecture, Engineering & Construction
University of Michigan
Plant Extension
Architecture, Engineering & Construction

Figure 5.2: The G.G. Brown building (above) and its as-built floor plan (below)

For the reasons mentioned before, it is highly recommended that owners inherit the as-built BIM of the building from the contractor upon a project's completion. However, as the BIM of the investigated building was not inherited by UM Plant Operations, it was created manually using the as-built HVAC system and floor plans of the facility. Figure 5.3 shows the HVAC system and the skin of the BIM corresponding to a section of the G.G. Brown building that was used to demonstrate the concepts, algorithms, and tools developed in this research.


GGBrown-with Ducts.dgn [3D - V8 DGN] - AECOsim Building Designer V8i (SELECTseries 3) (Licensed For Academic Use Only)

ACSPLANES loaded.

File Edit Element Settings Tools Utilities Workspace Building Designer Window Help
Compound
Part Level
Part Color
Part Style
Part Weigh
Building Designer Tasks
Architectural...
Structural Design
View 1 - Top, Design Model
View 2 - Right, Design Model
View 3, Design Model
View 4 - Front, Design Model
Design Model
None
Floor (ACS) Rotation 0.00°
Pan View > Select View
Default (none)

Figure 5.3: BIM showing the HVAC system of a section of the G.G. Brown building (Image courtesy Ali Golabchi)

The HVAC BIM is used to extract and store the distribution system logic as a tree network. This is accomplished by traversing all possible HVAC distribution paths—starting from the central HVAC unit and ending at the occupant rooms. HVAC components (such as supply ducts, heating/cooling coils, air blower units, etc.) are modeled as nodes and their physical connections/layout determines the tree network's precedence. The BIM of one such HVAC distribution network that supplies a portion of the G.G. Brown building is shown in Figure 5.4, and its corresponding tree network is illustrated in Figure 5.5. The problem of automatically detecting faults in the system then reduces to localizing the nodes that have most probably failed for a given set of reported complaints.

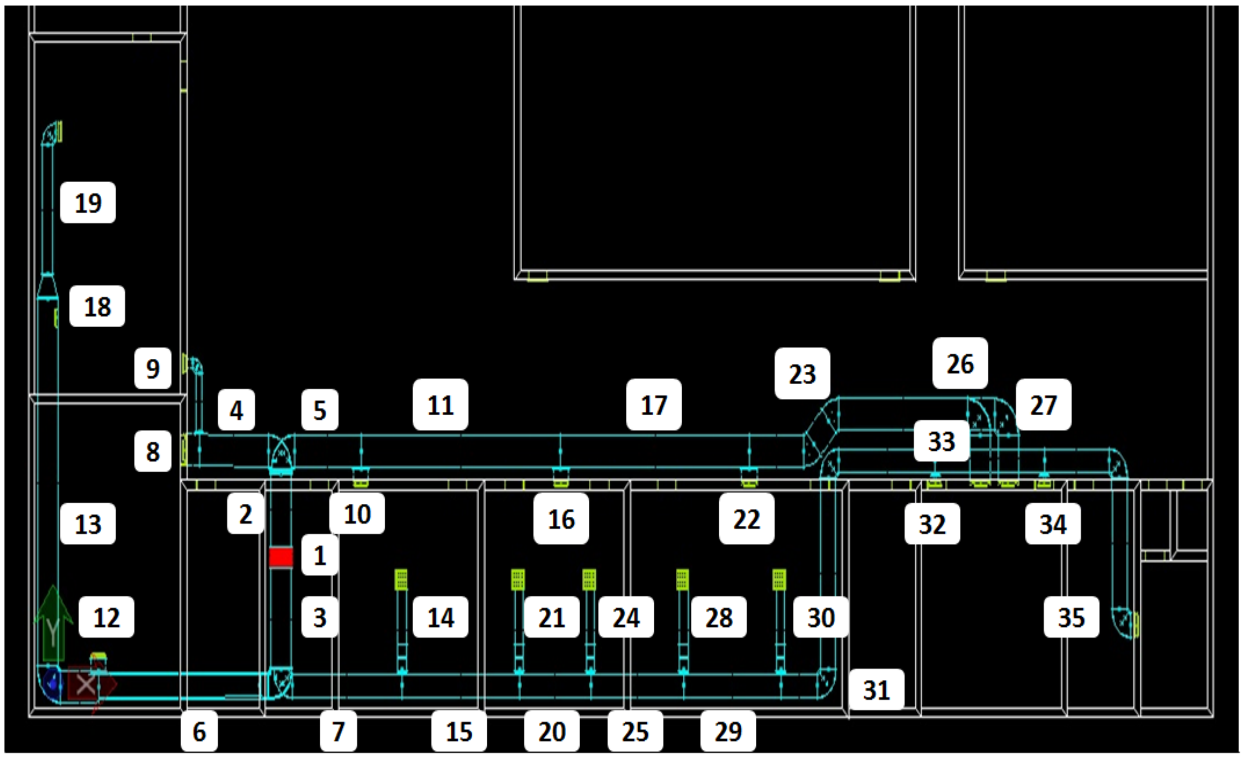


Figure 5.4: BIM of a HVAC distribution network supplying the south-west portion of G.G. Brown building

It is important to note that the fault detection method presented is limited by the use of occupant room temperatures, HVAC distribution network topology, and activity status as contextual parameters. The implementation of a fault detection algorithm that also considers the functional characteristics of the nodes in the HVAC distribution tree as contextual parameters is likely to improve the accuracy of the fault detection system.

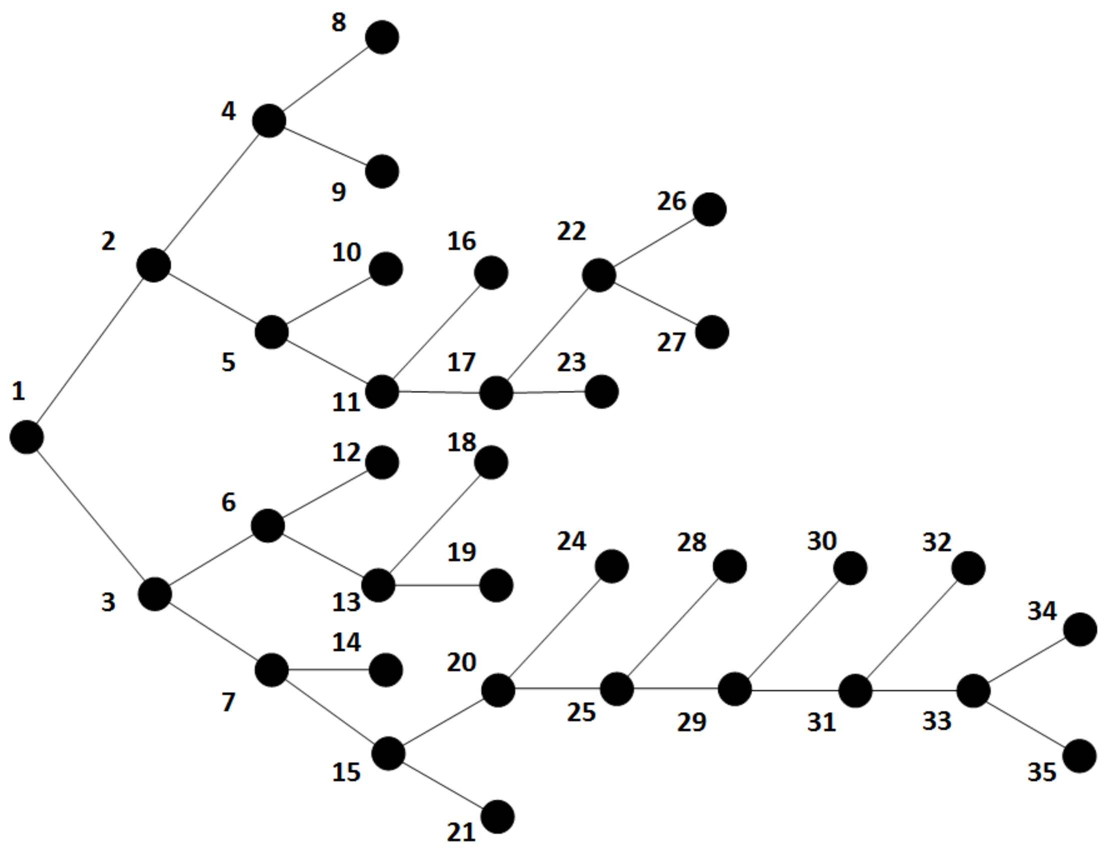
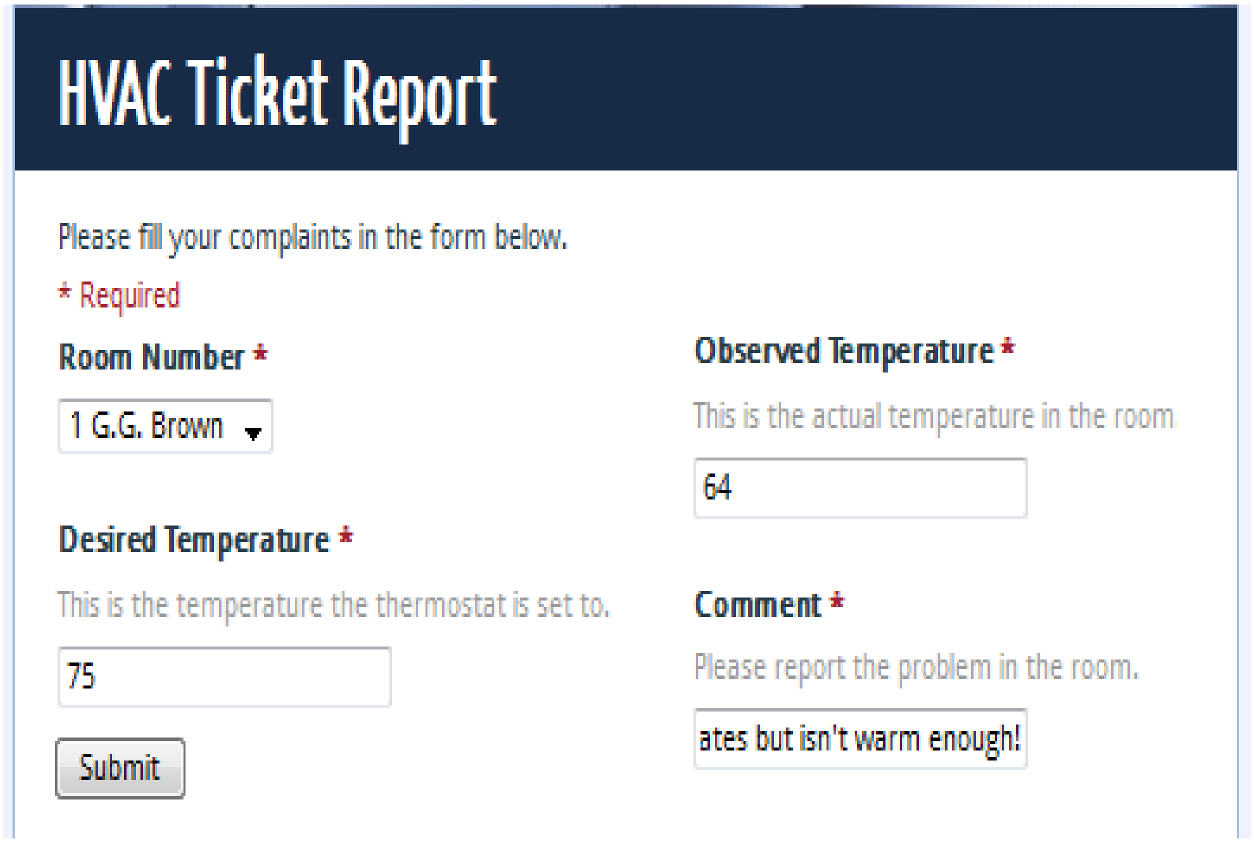


Figure 5.5: The tree network model of the HVAC distribution system

5.4.2 HVAC Malfunction Ticketing System

The web-form–based ticketing system shown in Figure 5.6, which can be accessed by all occupants, is used to register complaints about HVAC malfunction. Upon experiencing deteriorated performance, the occupant submits the information requested in the web-form, including room number, set (desired) temperature, and current (observed) temperature. The ticketing system also monitors and records the outdoor weather and temperature conditions as the complaint is submitted (Golabchi et al 2013). Temperature sensors installed in occupant rooms can also be used as a substitute for the manual ticketing system to record the performance of the HVAC system in the building.


HVAC Ticket Report

Please fill your complaints in the form below.

* Required

Room Number *

1 G.G. Brown

Desired Temperature *

This is the temperature the thermostat is set to.

75

Observed Temperature *

This is the actual temperature in the room

64

Comment *

Please report the problem in the room.

ates but isn't warm enough!

Figure 5.6: Web-form–based ticketing to record occupant complaints (Golabchi et al 2013)

5.4.3 HVAC Performance Deterioration Model

The deterioration in HVAC performance is modeled as shown in Figures 5.7 and 5.8 for heating and air conditioning, respectively. The outdoor, observed, and desired temperatures of the occupant's room are inherited from the ticketing system. The natural temperature of the room is the temperature of the room without any HVAC supply for a given outdoor temperature. The natural temperature is a function of the outdoor temperature and the materials used in constructing the facility and can be determined by recording a series of observations when the HVAC unit is turned off.

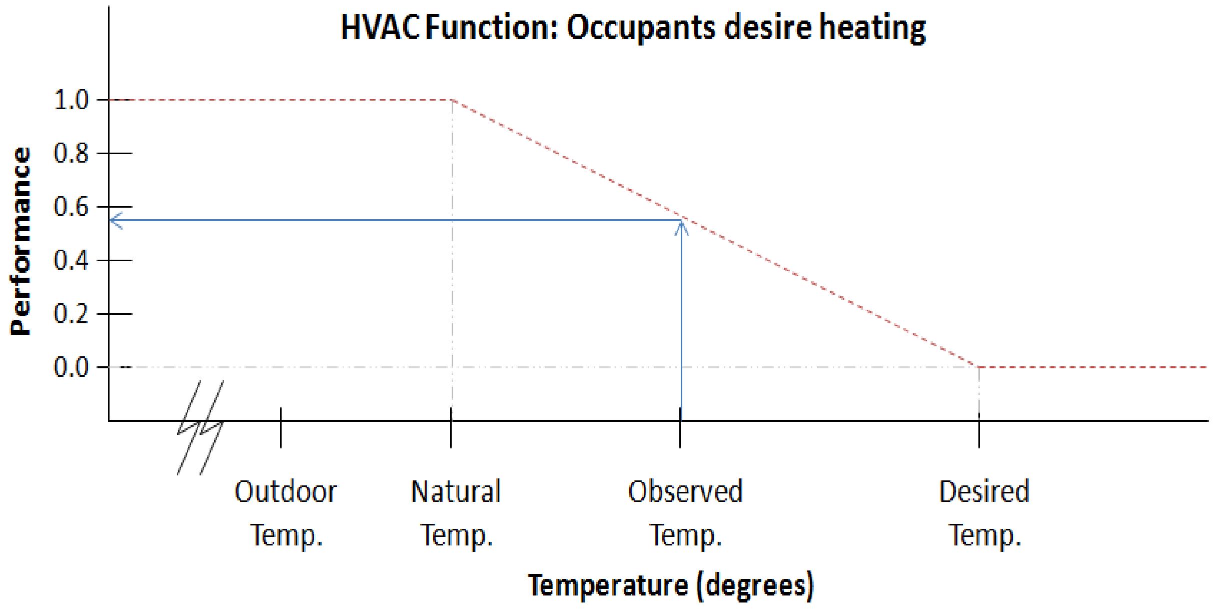


Figure 5.7: HVAC performance deterioration model for heating

The deterioration model maps the HVAC performance on a scale of 0 to 1, with 0 representing perfect performance and 1 representing total failure. If the observed temperature is equal to the desired temperature, it is considered a perfectly functioning system. If the observed temperature is equal to the natural temperature, it is considered a total failure of the HVAC system. The performance deterioration between the desired and natural temperatures is modeled as a linear function. The occupied rooms, where deterioration in performance may be reported, are the leaf nodes of the HVAC distribution tree network.

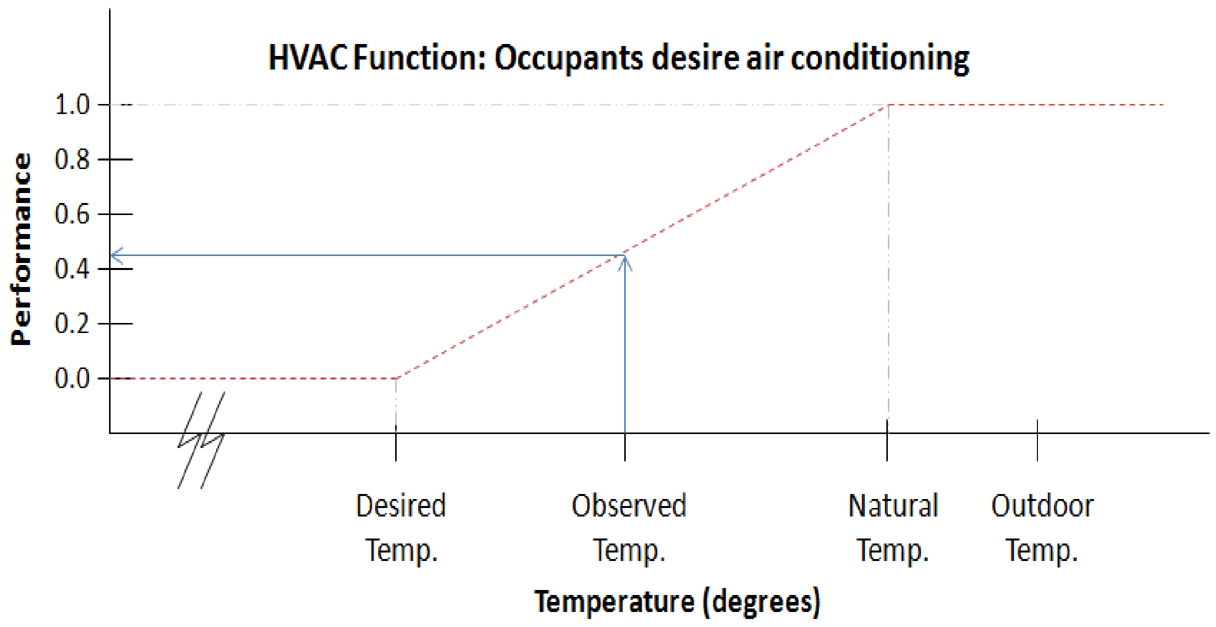


Figure 5.8: HVAC performance deterioration model for air conditioning

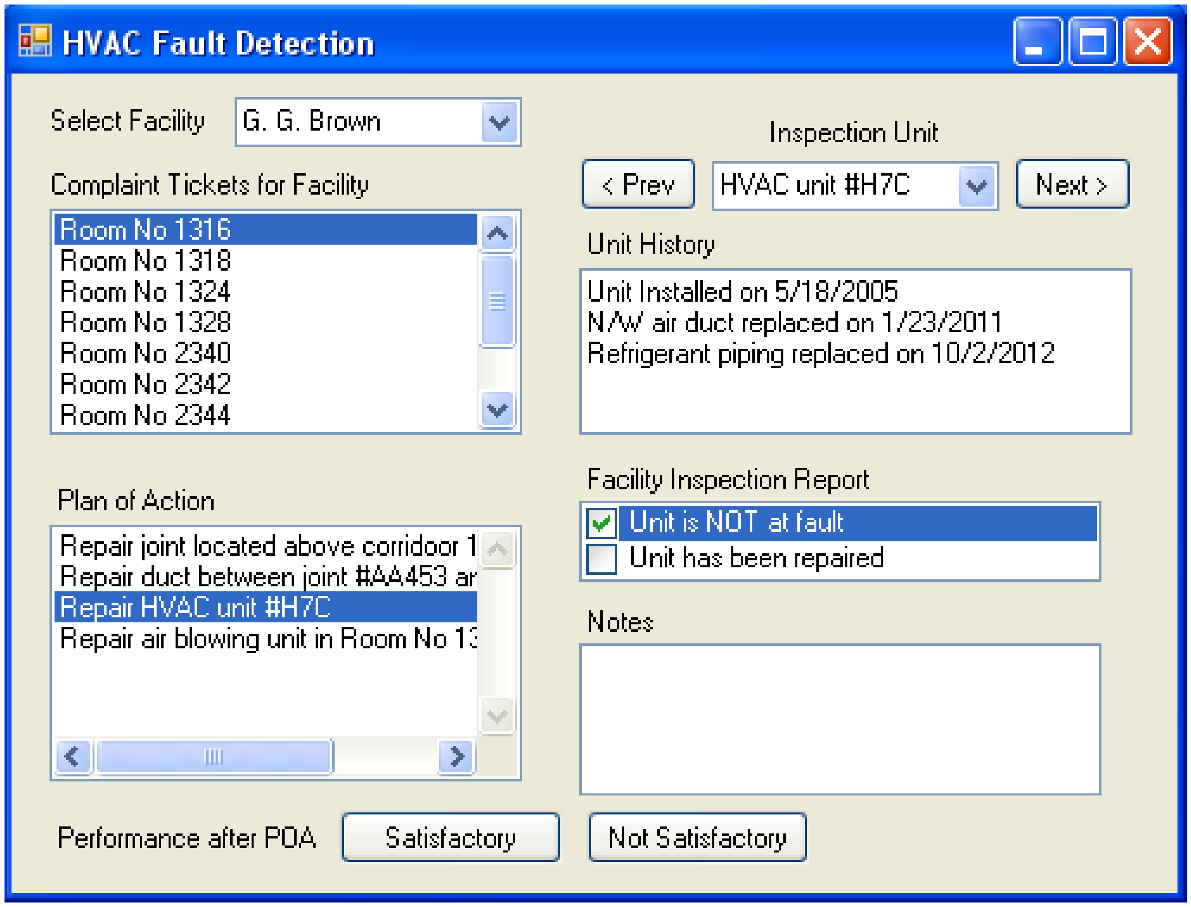
5.4.4 Fault Detection Algorithm

The author adopts an algorithm (Leckie and Dale 1997) that uses the information-theoretic minimum message length principle (Georgeff and Wallace 1984; Wallace and Boulton 1968) to locate faults in the HVAC tree-structure network model. The inputs of the algorithm are the HVAC distribution tree-network, the HVAC performance at the leaf nodes, and the probability distribution of HVAC performance at the leaf nodes for two cases—when the occupant room corresponding to the leaf node is affected by HVAC failure, and when it is unaffected. For each occupant room (leaf node), the probability density function of HVAC performance—if the occupied room is affected by HVAC failure—is modeled by observing the reported performance metrics over several historic instances of HVAC failure. The author assumes that if HVAC failure does not affect the occupant room, then the occupant will not report deteriorated performance, and the default performance of such leaf nodes is assumed to be 0 (perfect).

A set of observed HVAC leaf node performances could be the result of several possible fault combinations. For each such possible fault combination, the algorithm first constructs a message that explains the fault combinations and the performances. The algorithm then selects the fault combination with the minimum message length to be the most plausible fault combination (Leckie and Dale 1997). The algorithm localizes a set of HVAC components that are most likely at fault for the deteriorated performances observed in the system. A detailed discussion of the adopted minimum message length algorithm is presented in Appendix B.

5.4.5 HVAC System Repair Workflow

The author developed a BIM plug-in application, shown in Figure 5.9, to assist HVAC facility inspectors in determining their plan of action. The plug-in was written in C# using MicroStation API for seamless integration with MicroStation and Bentley BIM products. Upon querying, the application extracts and stores the HVAC distribution network models (corresponding to the facility) as tree data structures. The application acquires the tickets corresponding to the facility and generates leaf node performance measurements using the deterioration models. Next, the application invokes the fault detection algorithm and determines the set of HVAC components that are most likely at fault. The application then generates the plan of action by requesting that the inspector repair the components suspected to be at fault—starting from those farthest away from the root of the HVAC distribution tree.


HVAC Fault Detection

Select Facility: G. G. Brown

Complaint Tickets for Facility:
- Room No 1316
- Room No 1318
- Room No 1324
- Room No 1328
- Room No 2340
- Room No 2342
- Room No 2344

Inspection Unit: HVAC unit #H7C

Unit History:
Unit Installed on 5/18/2005
N/W air duct replaced on 1/23/2011
Refrigerant piping replaced on 10/2/2012

Plan of Action:
- Repair joint located above corridoor 1
- Repair duct between joint #AA453 ar
- Repair HVAC unit #H7C
- Repair air blowing unit in Room No 13

Facility Inspection Report:
- [✓] Unit is NOT at fault
- [ ] Unit has been repaired

Notes:

Figure 5.9: BIM plug-in for automated fault detection in HVAC system failure

For each component suspected to be at fault, the inspector either: 1) finds the equipment to be faulty, repairs it, and reports it as repaired, or 2) finds no fault in the equipment and reports it to be undamaged. After investigating all of the suspected components, the inspector returns to the rooms where deteriorated performance was recorded and monitors the HVAC performance. If the HVAC performance is not found to be satisfactory in some rooms, the inspector generates new tickets for all such rooms and cancels previously recorded tickets.

The plug-in then modifies the HVAC tree-network by eliminating those nodes corresponding to the list of components that were deemed to be undamaged and/or repaired by the inspector. The sub-trees originating at these nodes are attached to the node's parents. The plug-in then utilizes the newly generated HVAC tree-network and ticket data to determine a plan of action for the inspector to follow. This process is repeated until the HVAC performance is deemed satisfactory by the occupants. The author proposes using this BIM plug-in on a tablet device at the job site; this will highly reduce the time and effort spent by the FM personnel by eliminating the need to go through all the paper plans and blueprints at the main office.

5.4.6 Limitations of the Research

The author assumes that occupants experience either perfect or deteriorated HVAC performance and do not consider the rare cases in which HVAC over-performs. The HVAC performance deterioration is modeled as a linear mapping but in reality 'performance' is a subjective term and the deterioration model experienced by most occupants is not necessarily linear. The probability density function of HVAC performance, under failure, is developed by observing performance metrics corresponding to failure history. However, historic patterns of HVAC performance, in case of failure, may not be an accurate representation of future performance under similar conditions.

The accuracy of the fault detection algorithm in identifying the HVAC components at fault compared to the accuracy of manually doing the same (based on judgment, knowledge, and experience) is unknown and must be investigated. The fault detection algorithm only considers occupant room temperatures, HVAC distribution network topology, and activity status as contextual parameters. The implementation of an algorithm that also considers the functional characteristics of the nodes in the HVAC distribution tree as contextual parameters is likely to improve the performance of the fault detection system.

5.5 Summary and Conclusions

The use of BIM in the AEC industry is largely restricted to the conception, design, and construction phases of a building's lifecycle. In the O & M phase, BIM can be used by FM personnel as a database to document evolving facility information. The author believes that the true potential of BIM in FM is realized by using the BIM database to automate decision-making tasks.

The potential of using BIM to automate decision making by using the scenario of HVAC system failure has been demonstrated in this chapter. One method to model the HVAC distribution system and the deteriorated HVAC performance that are used by the fault detection algorithm is also presented. The author deployed a fault detection algorithm that uses the minimum message length principle to localize those HVAC components most likely at fault. Based on the fault detection algorithm, a workflow method is presented that uses the developed BIM plug-in to generate the facility inspector's plan of action.

The developed BIM plug-in guides the operation of HVAC repair and eliminates the time and effort spent by FM personnel to manually do the same based on their judgment and experience. The proposed automated fault detected application is best suited to be deployed using BIM software installed on a tablet device for FM field personnel's convenience. However, the development of BIM applications on tablet devices is in its early stages and should be investigated by the research community, since FM personnel can highly benefit from its advantages.

5.6 Acknowledgments

The author would like to thank Jerome Schulte and Mark Itil of UM Plant Operations for providing the HVAC plans used in this research. The author would also like to thank PhD student Ali Golabchi for his generous assistance in creating BIM models used to demonstrate the implementation of the fault detection algorithm and BIM tools developed as part of this research. Any opinions, findings, conclusions, and recommendations expressed in this chapter are those of the author and do not necessarily reflect the views of the individuals mentioned here.

5.7 References

Akcamete, A., Akinci B., and Garrett J.H. (2010). "Potential Utilization of Building Information Models for Planning Maintenance Activities," Proc. of the Int. Conference on Computing in Civil and Building Engineering, Nottingham, UK, pp. 151–158.

Anoop, S., Salman, A., and Joseph, T. (2011). "Preparing a Building Information Model for Facility Maintenance and Management," Proc. of the 28th Int. Symp. on Automation and Robotics in Construction (2011 ISARC), Seoul, Korea, pp. 357–375.

Azhar, S., Hein, M., and Sketo, B. (2008). "Building Information Modeling (BIM): Benefits, Risks and Challenges," Proceedings of the 44th Associated Schools of Construction (ASC) Annual Conference, Auburn, Alabama, pp. 2–5.

Barnes, S., and Castro-Lacouture, D. (2009). "BIM-Enabled Integrated Optimization Tool for LEED Decisions," Proc. of 2009 ASCE Int. Workshop on Computing in Civil Engineering, Technical Council on Computing and Information Technology of ASCE, American Society of Civil Engineers (ASCE), Reston, VA, pp. 258–268.

Becerik-Gerber, B., Jazizadeh, F., Li, N., and Calis, G. (2012). "Application Areas and Data Requirements for BIM-Enabled Facilities Management," Journal of Construction Engineering and Management, American Society of Civil Engineers (ASCE), Vol. 138(3), pp. 431–442.

BuildingSMART Alliance (2013). "buildingSMART Alliance" – Website of the National Institute of Building Sciences <http://www.buildingsmartalliance.org/index.php/nbims/faq/> as viewed on 14th January, 2013.

Cho, Y.K., Alaskar, S., and Bode, T.A. (2010). "BIM-Integrated Sustainable Material and Renewable Energy Simulation," Proc. of Construction Research Congress 2010: Innovation for Reshaping Construction Practice, Construction Institute of ASCE, American Society of Civil Engineers (ASCE), Reston, VA, pp. 288–297.

Dunston, P.S., and Williamson, C.E. (1999). “Incorporating Maintainability in Constructability Review Process,” Journal of Management in Engineering, American Society of Civil Engineers (ASCE), Vol. 15(5), pp. 56–60.

East, W., and Brodt, W. (2007). "BIM for Construction Handover,” Journal of Building Information Modeling, National BIM Standard and National Institute of Building Sciences, Matrix Group Publishing, Winnipeg, Canada, pp. 28–35.

Eastman, C., Teicholz, P., Rafael, S., and Kathleen, L. (2008). “BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors,” John Wiley and Sons Inc., New Jersey, NJ.

Georgeff, M.P., and Wallace, C.S. (1984). “A General Criterion for Inductive Inference,” Proc. of the 6th European Conference on Artificial Intelligence (ECAI-84), European Coordinating Committee for Artificial Intelligence (ECCAI), Pisa, Italy, pp. 473–482.

Goedert, J. and Meadati, P. (2008). “Integrating Construction Process Documentation into Building Information Modeling,” Journal of Construction Engineering and Management, American Society of Civil Engineers (ASCE), Vol. 134(7), pp. 509–516.

Golabchi, A., Akula, M., and Kamat, V.R. (2013). “Leveraging BIM for Automated Fault Detection in Operational Buildings,” Proc. of the 30th Int. Symp. on Automation and Robotics in Construction (2013 ISARC), Montreal, Canada (In Press).

Huber, D., Akinci, B., Oliver, A.A., Anil, E., Okorn, B.E., and Xiong, X. (2011). "Methods for Automatically Modeling and Representing As-built Building Information Models," Proc. of the NSF Engineering Research and Innovation Conference, National Science Foundation (NSF), Atlanta, GA.

Hwang, S., and Liu, L. (2010). "BIM for Integration of Automated Real Time Project Control Systems," Proc. of Construction Research Congress 2010: Innovation for Reshaping Construction Practice, Construction Institute of ASCE, American Society of Civil Engineers (ASCE), Reston, VA, pp. 509–517.

Khemlani, L. (2009). "Sutter Medical Center Castro Valley: Case study of an IPD project," Website of AECbytes: Building the Future Article dated March 6th, 2009 accessible at <http://www.aecbytes.com/buildingthefuture/2009/Sutter_IPDCaseStudy.html> as viewed on 6th April, 2013.

Leckie, C. and Dale, M. (1997). "Locating Faults in Tree-Structured Networks," Proc. of the 15th Int. Joint Conference on Artificial Intelligence (IJCAI), Nagoya, Japan, pp. 434–439.

Leite, F., Akinci, B., and Garrett, J. (2009). "Identification of Data Items Needed for Automatic Clash Detection in MEP Design Coordination," Proc. of 2009 Construction Research Congress (2009 CRC): Building a Sustainable Future, American Society of Civil Engineers (ASCE), Reston, VA, pp. 416–425.

Liu, X., Eybpoosh, M., and Akinci, B. (2012). "Developing As-built Building Information Model Using Construction Process History Captured by a Laser Scanner and a Camera," Proc. of the 2012 Construction Research Congress (2012 CRC), West Lafayette, IN.

Madritsch, T., and May, M. (2009). "Successful IT Implementation in Facility Management," Facilities, Emerald Group Publishing, Bingley, UK, Vol. 27(11/12), pp. 429–444.

Motawa, I., and Almarshad, A. (2012). "A Knowledge-Based BIM System for Building Maintenance," Automation in Construction, Elsevier Science, NY, Vol. 29, pp. 173–182.

Newton, R. (2004). "Inadequate Interoperability in Construction Wastes 415.8 Billion Annually," AECNews.com, 13, Article 342.

Seul-Ki, L., Hyo-Kyung, A., and Jung-Ho, Y. (2012). "An Extension of the Technology Acceptance Model for BIM-based FM," Proc. of the 2012 Construction Research Congress (2012 CRC), West Lafayette, IN, pp. 602–611.

Singh, V., Gu, N., and Wang, X. (2011). "A Theoretical Framework of a BIM-based Multi-Disciplinary Collaboration Platform." Automation in Construction, Elsevier Science, NY, Vol. 20(2), pp. 134–144.

Smith, D. K., and Tardif, M. (2009). “Building Information Modeling: A Strategic Implementation Guide for Architects, Engineers, Constructors, and Real Estate Asset Managers,” John Wiley and Sons, Hoboken, NJ.

Stumpf, A., Kim, H., and Jenicek, E. (2009). “Early Design Energy Analysis using BIMs (Building Information Models),” Proc. of 2009 Construction Research Congress — Building a Sustainable Future, American Society of Civil Engineers (ASCE), Reston, VA, pp. 426–436.

Tatum, C.B., and Korman, T. (2000). “Coordinating Building Systems: Process and Knowledge,” Journal of Architectural Engineering, American Society of Civil Engineers (ASCE), Vol. 6(4), pp. 116–121.

Teicholz, E. (2004). "Bridging the AEC Technology Gap,” IFMA Facility Management Journal, International Facility Management Association (IFMA), Vol. 2.

Wallace, C.S., and Boulton, D.M. (1968). “An Information Measure for Classification,” Computer Journal, British Computer Society (BCS), Vol. 11(2), pp. 185–194.

Woo, J., Wilsmann, J., and Kang, D. (2010). “Use of As-Built Building Information Modeling,” Proc. of the 2010 Construction Research Congress (2010 CRC), American Society of Civil Engineers (ASCE), Banff, Canada.

Yu, K., Thomas, F., and Francois, G. (2000). "A Development Framework for Data Models for Computer-Integrated Facilities Management," Automation in construction, Elsevier Science, NY, Vol. 9(2), pp. 145–167.

Chapter 6

Context-Aware Framework for Highway Bridge Inspections

6.1 Introduction

There are 605,102 public highway bridges (that are 20 feet in length or longer) in the United States, as documented in the National Bridge Inventory, that are subject to the National Bridge Inspection Standards (NBIS) according to the Bureau of Transportation Statistics (Moore 2011). These bridges are typically under the jurisdiction of the Departments of Transportation (DOT) in the states where they are located. The Michigan Department of Transportation (MDOT), for example, lists 4,397 highway bridges under its jurisdiction—of which 341 are structurally deficient and 899 are functionally obsolete. The long-term deterioration and rehabilitation of bridges is an important problem that could lead to large-scale disasters if left unchecked.

6.1.1 Current State of Bridge Inspection Routines

States DOTs are generally responsible for the regular inspection and maintenance of the bridges under their jurisdiction. MDOT employs more than 20 full-time inspectors who have specialized training and who often work in teams of two. MDOT inspectors regularly evaluate bridges and assign condition ratings using the National Bridge Inspection Rating Scale based on NBIS (MDOT 2011; Farrar 2008). The complexity of the tasks and the scale of operations in this domain have significant potential for improvement as far as assessment accuracy, inspection productivity, and efficiency are concerned.

Traditionally, bridge inspections are documented manually—the bridge inspector visually assesses the condition of a bridge based on standard rating guidelines and previous bridge inspection reports (Farrar 2008). While conducting an inspection, the inspector carries rating guidelines and previous reports in their paper form along with the current report form. In many cases, the inspector has to come to the site with significant preparation, and plenty of time and effort is consumed in searching, organizing, and retrieving relevant information. Upon completing the inspection and returning to the office, the relevant data gathered on the field is entered into a database management system.

In recent years, researchers have suggested methods to assess infrastructure condition by using IT tools including automatic extraction of deterioration characteristics from point clouds (Atasoy et al 2010; Tang and Akinci 2012) and video imaging data (Koch and Brilakis 2011). The use of bridge inspection reporting software has been explored by several state DOTs—such as Indiana's INDOT (Indiana State BIAS 2012), Wisconsin's WisDOT (Adams et al 2005), and Washington's WSDOT (BridgeWorks.NET 2012)—as well as by independent asset management software developers (BridgeInspect^{TM} 2012). These solutions incorporate a centralized database and software installed on the inspector's tablet-PC to facilitate the inspection process in the field. The bridge inspection software consists of interactive forms corresponding to specific components of the bridge being assessed. These forms can retrieve customized inspection guidelines and relevant historic bridge inspection data, capture bridge evaluation data, and automatically associate the captured information with the bridge components making the bridge inspection documentation intuitive.

During an inspection routine, the inspector visually assesses the condition of a particular bridge component. Based on a priori knowledge of the bridge components' taxonomic hierarchy and ontology, the inspector navigates to the form corresponding to the component. The inspector then reports the component's condition based on the visual assessment and the aforementioned features incorporated in the software. The component's assessment is documented in the centralized database and the inspector continues this process with the remaining bridge components. The process of navigating to the appropriate form corresponding to the component being assessed can, however, be time consuming, and it requires a priori knowledge of the vast domain of bridge components' taxonomic hierarchy and ontology across several bridges. Context-aware computing has the potential to address these shortcomings and challenges by automatically identifying the components in an inspector's context.

6.1.2 Introduction to Context-Aware Computing

Context-aware computing helps deliver relevant information to on-site mobile users by interpreting the user's context—defined by using environmental characteristics such as the user's location, time, identity, profile, and activity (Burrell and Gay 2001). By leveraging knowledge about various contextual parameters, context-aware computing can deliver highly specific information pertinent to the decisions at hand and can thus be of great value for civil engineering inspectors, emergency responders, facility managers, and maintenance crews. Context-aware computing applications are implemented using a mobile distributed computing system—a collection of mobile and stationary sensing and computing devices. These applications examine and react to the mobile user's changing context in a mobile, dynamic (indoor and outdoor) computational environment (Schilit et al 1994).

As mentioned previously, context-aware computing has the potential to address the shortcomings and challenges of navigating the bridge inspection report forms by automatically identifying the bridge components in the inspector's context. Assisted by position tracking devices that are strategically placed in the inspector's body suit, the spatial profile parameters of the inspector are acquired in real-time. Based on the acquired profile parameters, context-aware computing can identify bridge components relevant to the inspector's current context. Bridge inspection reporting software can be integrated with context-aware computing to automate the process of navigating and populating the appropriate report forms based on the components identified to be in context. Supplementing field inspectors with relevant contextual data to support their operations can thus potentially improve the efficiency of bridge inspection routines.

6.1.3 Overview of the Chapter

The objective of this chapter is to present the research that investigated potential context-aware computing methods that can be integrated with bridge inspection routines. This chapter first presents a context-aware computing framework developed for the automatic identification of contextual bridge components. As part of the overarching framework, the chapter presents a point cloud–based method to represent the geometry of bridge components' hierarchy. Next, a method to geometrically construct the inspector's region of view as a truncated cone centered about the line of sight is presented. An algorithm to reconcile the points contained within a user's viewing cone to hierarchical bridge entities—in order to determine contextual bridge components—is then presented. The chapter then covers an algorithm to prioritize contextual components in order to identify the specific component considered to be the most relevant to the user's operational and spatial context.

The author presents potential technologies that can be used to track the location and head orientation (i.e., the pose) of the bridge inspector. The uncertainty in position tracking technologies results in an inaccurate interpretation of the spatial context of the inspector, which leads to errors in the components identified to be within spatial context. Using a case study, the chapter then presents the sensitivity analysis that evaluates the framework performance and illustrates the effect of uncertainty in tracking technologies on the accuracy of determining contextual components. The feasibility of using Global Positioning System (GPS) and magnetic compass for tracking contextual parameters is evaluated. Finally, the author presents a discussion on the challenges and implications of integrating context-aware computing into bridge inspection applications and implementing them in an outdoor, unconfined, and spatially expansive environment.

6.2 Current State of Knowledge

Context-aware systems are concerned with three major features: the acquisition of context, the abstraction and understanding of the context, and the reaction of the application to the recognized context (Schmidt 2002). Spatial context is one of the most important parameters considered in developing context-aware computing applications. In information-intensive industries, it is crucial to critical decision-making that there be rapid, convenient, and real-time access to the spatial context of workflow entities.

Currently, there is a proliferation of spatially aware applications in healthcare (Bardram and Christensen 2007; Jahnke et al 2005), vehicle intelligence and navigation technology (Taylor et al 2001; Bachmann et al 2000; Hoch et al 2007), and mobile tourism in environments characterized by high information density (Aittola et al 2003; Davies et al 1999; Fleck et al 2002; Laukkanen et al 2002; Schwinger at al 2009). The recent development of "smartphone" and computing-tablet technology has provided civilian population with malleable personal digital assistants (PDAs) that have localization features based on GPS, wireless cellular networks, and accelerometers. This has led to the development of several popular spatially-aware applications customized for PDAs.

6.2.1 Context-Aware Computing in the Civil Infrastructure Industry

In the construction industry, field personnel spend a significant amount of time manually accessing assets and information required to support construction workflow tasks, leading to a loss in productivity and money (Aziz et al 2005; Stukhart and Bell 1985; The Business Roundtable 1982). It has been estimated that as much as 30% of project costs are wasted in the construction industry (Zou et al 2006) due to: 1) unreliable documentation, access, and delivery, 2) ineffective communication, and 3) inefficient operation organization (Fathi et al 2006). Real-time location sensing has been used to develop workforce, equipment, material, and information tracking for civil infrastructure applications. Such spatially aware applications have been deployed by installing jobsites with wireless local area networking (WLAN) communication

(Aziz et al 2006), digital radio–based ZigBee networks (Skibniewski and Jang 2006) and Ultra-Wide Band (UWB) wireless sensing technology (Teizer et al 2009). In the aforementioned efforts, spatial context is usually defined solely by the location of the workflow entity. While this is an acceptable definition for entities such as equipment, material, and information, defining the spatial context of a mobile human user solely by the location leads to an incomplete and inaccurate interpretation of the user's spatial context.

A high-precision context-aware computing framework has been developed incorporating the mobile user's three-dimensional head orientation to interpret the user's fully qualified pose and spatial context (Khoury and Kamat 2009a). The overview of the high-precision context-aware computing framework is shown in Figure 6.1, using a potential bridge inspection scenario for illustration.

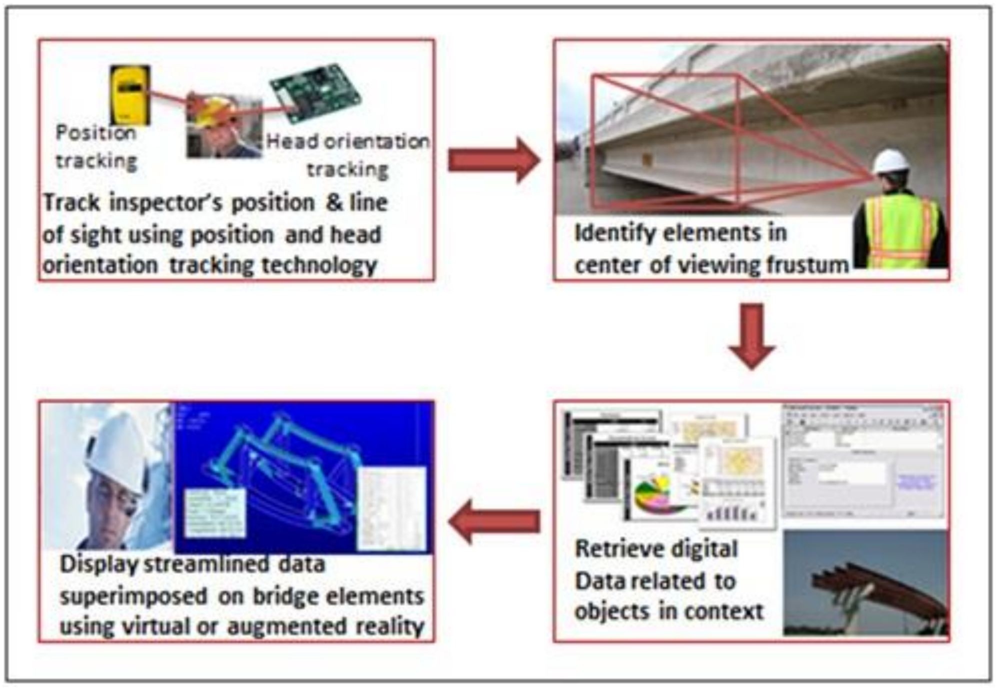

Position tracking
Head orientation tracking
Track inspector's position & line of sight using position and head orientation tracking technology
Identify elements in center of viewing frustum
Retrieve digital Data related to objects in context
Display streamlined data superimposed on bridge elements using virtual or augmented reality

Figure 6.1: Overview of high-precision context-aware information retrieval methodology

The pose of the mobile inspector is tracked to geometrically determine the region of space visible to the inspector at a particular instant. The contextual bridge components are then determined by computing the list of components whose geometry intersects with the geometry of the viewing region. The framework retrieves digital information relevant to the identified contextual components and this streamlined data is then displayed to the inspector in order to help them make informed decisions for completing the task at hand.

Khoury and Kamat (2009a) explored the conceptual idea of context-aware computing for mobile construction applications primarily in indoor environments—with WLAN, UWB, and indoor GPS (iGPS) localization technologies—and very limited testing was done in an outdoor environment. The scale and distribution of the objects in the considered environments were such that, with sufficient proximity, it was clearly identifiable which specific object a mobile user was interested in interacting with. Compared with this prior work, this chapter advances the discussed ideas by studying the technical and practical challenges of designing and implementing a context-aware engineering application in an outdoor, unconfined, and spatially expansive environment, such as a highway bridge. This chapter also conducts a detailed statistical analysis to document and demonstrate—to bridge inspectors as well as other mobile context-aware computing users—the sensitivity of the involved spatial context tracking parameters and the effect of their uncertainty on objects identified (or not) as being contextual. The research presented evaluates the feasibility of using GPS and a magnetic compass for outdoor localization and orientation tracking in practical context-aware applications.

6.3 Context-Aware Computing Framework

6.3.1 Integrating Context-Aware Computing with Bridge Inspection Reporting Applications

As mentioned previously, several IT solutions for bridge inspection reporting incorporate a centralized inspection database and reporting software installed on the inspector's tablet-PC to facilitate the inspection process in the field. The inspector manually navigates through the reporting forms and selects the appropriate form corresponding to the component in context. Upon selection, the reporting software queries the inspection database, retrieves streamlined information regarding the component, and displays the information to the inspector in a consumable format. Context-aware computing can be integrated with bridge inspection routines to automate the querying process, as shown in Figure 6.2.

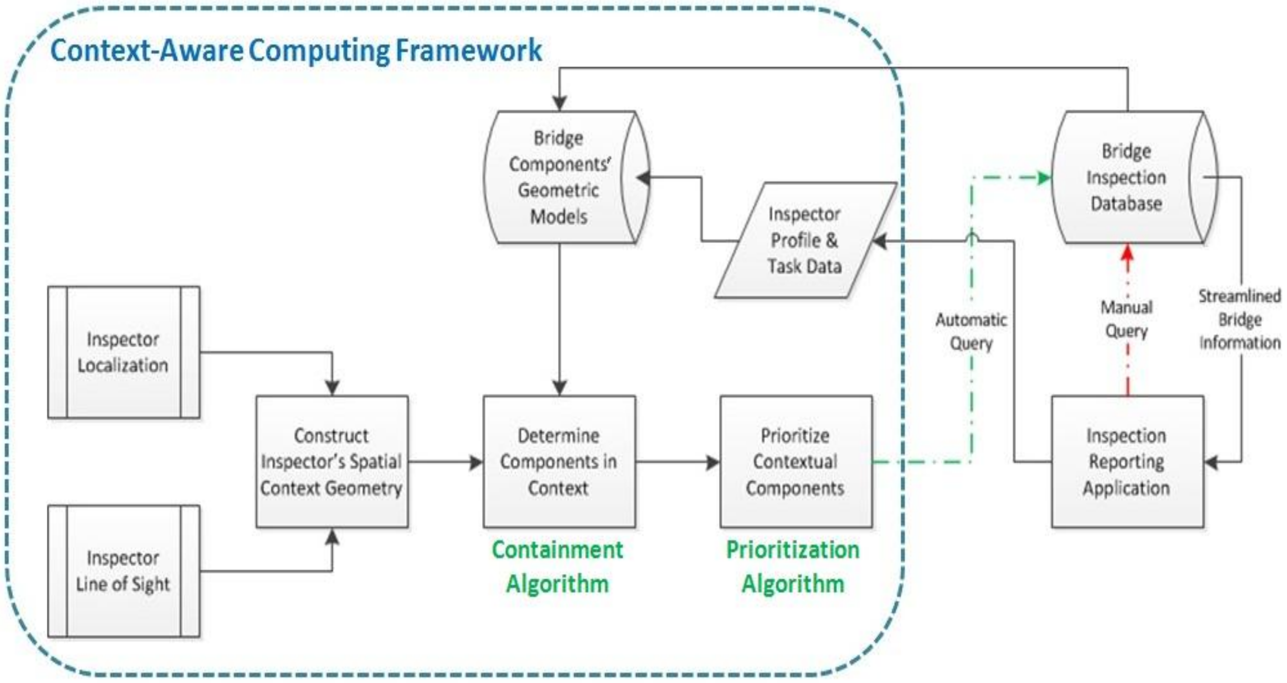

Context-Aware Computing Framework
Bridge Components' Geometric Models
Inspector Profile & Task Data
Bridge Inspection Database
Inspector Localization
Construct Inspector's Spatial Context Geometry
Determine Components in Context
Prioritize Contextual Components
Inspection Reporting Application
Inspector Line of Sight
Automatic Query
Manual Query
Streamlined Bridge Information
Containment Algorithm
Prioritization Algorithm

Figure 6.2: Context-aware computing framework integrated within the bridge inspection application architecture

The bridge inspection database is extended to include the geometric models of the bridge components. The framework tracks the pose of the mobile inspector and constructs a geometric model of the spatial region in the inspector's vision. The framework then co-ordinates the geometry of the bridge components with the constructed viewing region to compute the list of components within the inspector's spatial context. The contextual components are then prioritized to determine the component of interest and automatically generate the query. Automating the querying process can potentially save the time spent in manually navigating the software and reduces the effort spent by inspectors to learn, remember, and recall the taxonomic hierarchy and ontology of bridge components across a large fleet of bridges.

The framework was designed, implemented, and evaluated using the I-275 highway bridge over Telegraph Road (US-24) in Monroe, Michigan as the case study bridge (CSB). The CSB was recommended by MDOT for two primary reasons: 1) it is the best topological and ontological representative of most highway bridges in the state of Michigan; and 2) it is comprised of a significantly diverse variety of components found in a typical Michigan highway bridge.

6.3.2 Tracking Inspector Location and Line of Sight

The ability to ubiquitously track a mobile inspector's position and line of sight continuously and accurately is of the utmost importance as it establishes the parameters to construct the inspector's region of view. GPS is a convenient option to track a mobile user continuously in an outdoor environment (Hofmann-Wellenhof et al 1993). The GPS localization system is a space-based global navigation satellite system that provides reliable location information in all weather at all times and anywhere on earth where there is an unobstructed line of sight to four or more GPS satellites.

Location tracking applications based on GPS are available at several levels based on the accuracy required by the users. The author selected Real-Time Kinematic GPS (RTK-GPS) and Wide Area Augmentation System-capable GPS (WAAS GPS) for location tracking in GPS available environments. RTK satellite navigation is a technique used in land survey and in hydrographic survey based on the use of carrier phase measurements of the GPS where a single reference station provides the real-time corrections, providing up to centimeter-level accuracy. WAAS is a navigation aid service developed by the Federal Aviation Administration; it uses a network of ground-based reference stations and provides position corrections to GPS receivers to improve their accuracy.

The mobile inspector is fitted with a body suit that ergonomically holds the GPS tracking system hardware, as shown in Figure 6.3. The GPS receiver unit is fixed on top of the inspector's helmet and is connected to the computation center, which is a tablet-PC, which can either be placed in the inspector's vest or can be held by the inspector. The typical nominal uncertainty for RTK-GPS is 5 cm horizontally and 10 cm vertically. The typical nominal uncertainty for WAAS GPS is about 3 m.

When the inspector moves into a GPS-denied environment, such as the underside of the bridge, the inspector must rely on a non-GPS based tracking technology. Several localization systems, such as WLAN-based, UWB-based, and iGPS-based systems, have been developed and are commercially available for localization in GPS-denied environments.

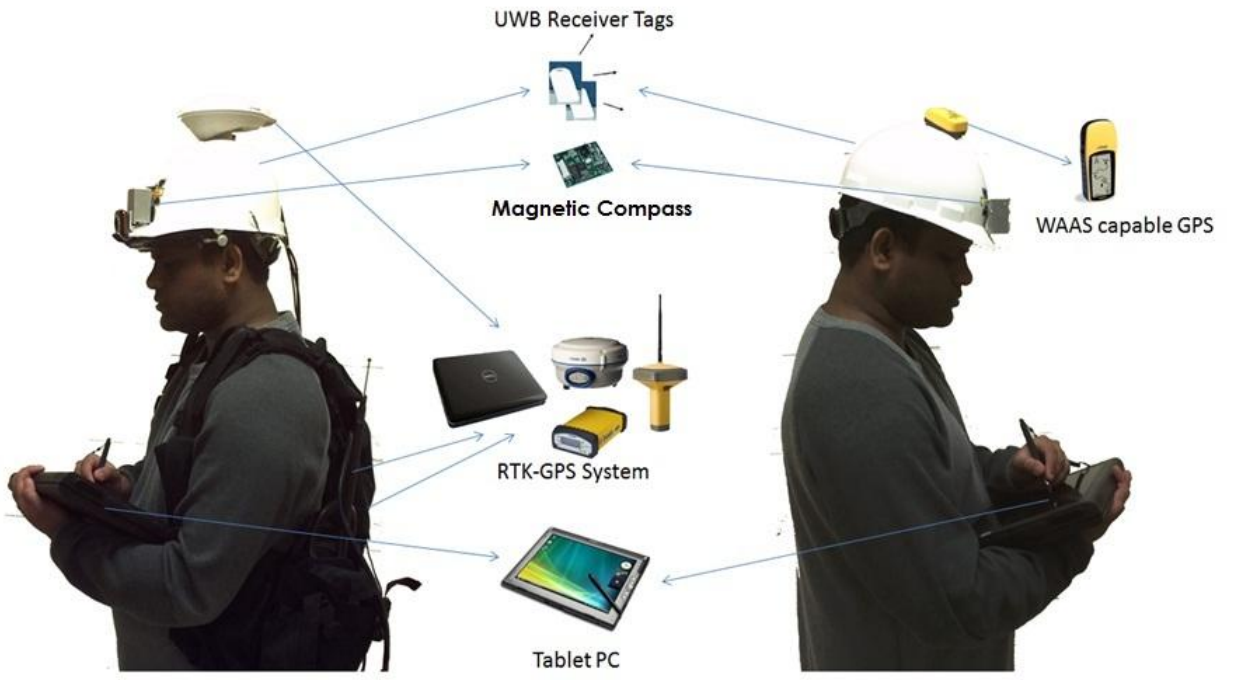


Figure 6.3: The body suit architecture incorporating position tracking system hardware

A detailed comparison of the WLAN-based, UWB-based, and iGPS-based systems has also been done in a recent study (Khoury and Kamat 2009b) and is summarized in Table 6.1. The author suggests the selection of a UWB-based localization system for location tracking in GPS-denied environments because it does not require calibration and has relatively low uncertainty (Multispectral Solutions Inc. 2012). The system uses UWB technology to determine the location of UWB Radio Frequency Identification (RFID) tags. The RFID tags are attached to the body suit of the mobile inspector, as shown in Figure 6.3. The tag repeatedly sends out bursts of UWB pulses that are received by UWB system receivers installed at known locations, typically around the periphery of the coverage area. Reception by three receivers allows for two-dimensional localization, while reception by four or more receivers allows for three-dimensional localization. The uncertainty for the UWB localization system was found to be within 0 cm to 50 cm (Khoury and Kamat 2009b).

	Line of sight	**Uncertainty**	**Calibration**	**Deployment and cost**
WLAN	Not required	1.5 m – 2 m	Needed (time consuming)	Easy and economical
UWB	Required (receiver-reference tag)	0 – 50 cm	Not needed	Quite easy but expensive
Indoor GPS	Required (receiver – transmitter)	1 – 2 cm	Needed (few sampling points)	Quite easy but very expensive

Table 6.1: Comparative summary of WLAN-based, UWB-based, and iGPS-based localization

The author selected magnetic orientation tracking for monitoring the inspector's line of sight. The tracking unit includes a built-in compass, and employs solid-state magnetic field sensors that measure compass heading through a full 360 degrees of rotation. The tracker employs proprietary hard and soft iron correction algorithms to calibrate out magnetic anomalies for repeatable, high-resolution measurement in challenging environments. The tracking unit is enclosed in an aluminum case that is rigidly mounted on the inspector's hard hat, as shown in Figure 6.3. Upon encountering a component of interest, the inspectors aligns the line of sight along the head orientation and face the component before communicating to the computing device to lock their pose. The magnetic compass calculates the inspector's head orientation, which is reconciled to estimate the line of sight direction in terms of roll, pitch, and yaw. The manufacturer-specified uncertainty of the orientation tracker is around 0.3° (PNI Sensor Corporation 2012).

6.3.3 Computing the Inspector's Spatial Context

Human vision is a complex phenomenon, making spatial field of view difficult to model. The approximate field of view of a human eye is 95° out, 75° down, 60° in, and 60° up (Grabe 2010), as illustrated in Figure 6.4. To reduce the complexity in order to spatially model the field of view, computer science literature, computer vision applications, video games, and Head Mounted Displays (HMD) model human vision as a truncated pyramid or a truncated cone and typically use a field of view of 45° in all directions (Rash et al 2009; Thomas et al 2000).

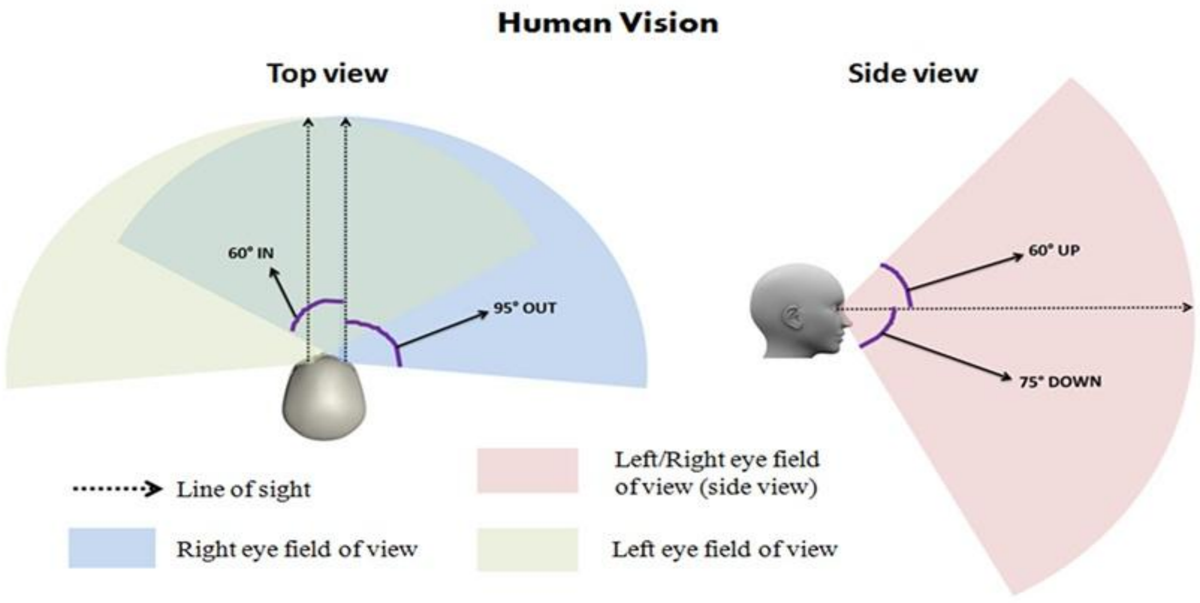


Figure 6.4: The top and side views of a human being's region of vision

The location and line of sight direction of the inspector are used to define the line of sight vector, and the region of space that represents the inspector's field of view is computed, as shown in Figure 6.5. The framework models the region of space that is of interest to the mobile inspector as a conical frustum whose axis is the line of sight of the mobile user, and whose a half angle is equal to the uniform field of view angle. The conical frustum is truncated between two planes perpendicular to the line of sight—the near and far planes. Objects closer to the inspector than the near plane and beyond the far plane of view are considered to be out of sight and thus out of context. The near and far plane distances are set to be 0 m and 30 m respectively.

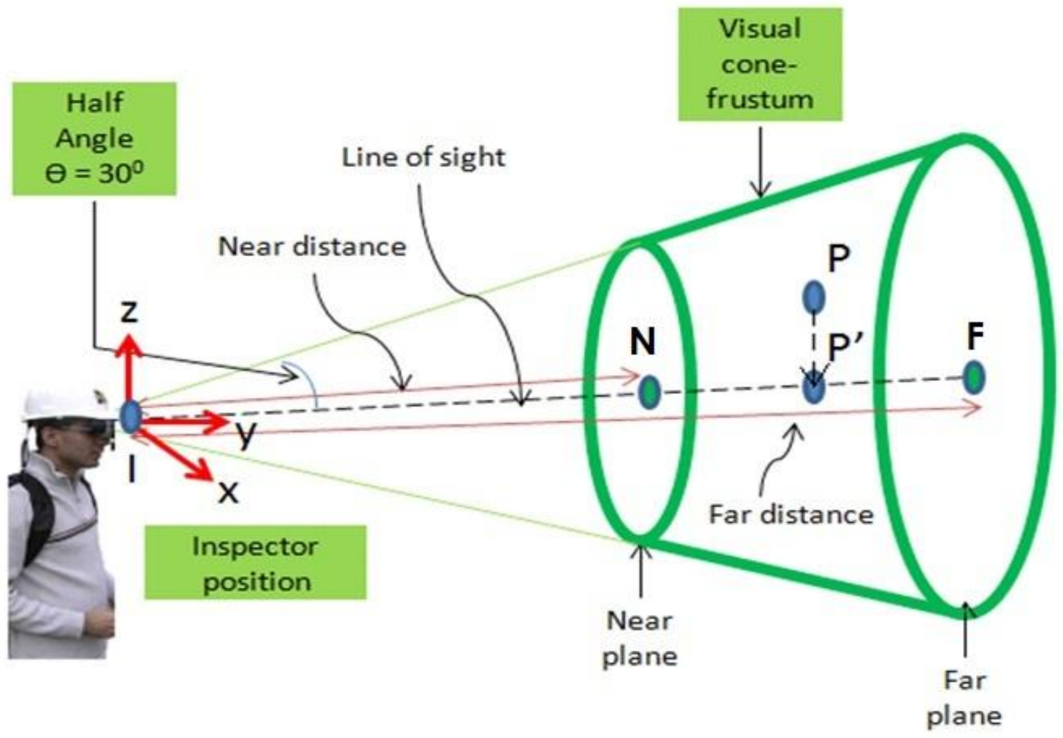


Figure 6.5: A geometric representation of the mobile inspector's viewing frustum

6.3.4 Geometry of Bridge Components

The CSB is decomposed into constituent components based on the Pontis bridge component schema (AASHTOWare 2012), which are listed in Table 6.2. The CSB, shown in Figure 6.6 (a), is composed of 62 components belonging to four component groups: deck, superstructure, substructure, and approach. The geometry of the components can be stored in several formats—such as CAD models, point clouds, and polygon soups—depending on the containment algorithm used by the context-aware computing framework. In this research, the components are modeled as a point cloud because the framework utilizes a point containment algorithm to determine contextual components.

Component Group	NBIS Structural Element Group	Pontis Components
Deck	Yes	3 deck surfaces, 6 deck railings, 6 deck sidewalks, 1 expansion joint, and 3 other joint components
Superstructure	Yes	14 bearings and 7 girder components
Substructure	Yes	2 abutments, 8 piers, and 2 slope protection components
Approach	No	2 approach pavements, 4 approach sidewalks, and 4 approach slope components

Table 6.2: The list of CSB bridge components according to the Pontis schema

The point cloud was developed using a CAD model of the bridge in which the constituent components are modeled as geometric primitives—cuboids, spheres, cylinders, pyramids, and prisms. The point cloud is created by generating points on the desired surfaces. Each point in the generated point cloud is mapped to the component it geometrically represents along with its coordinates. For example, in the CAD model of the CSB shown in Figure 6.6 (b), the deck surface component is modeled as a parallelepiped geometric primitive. The geometric equations of the parallelogram planes (the surfaces visible to the inspector) of the parallelepiped primitive are then extracted. The points corresponding to this component are generated using these geometric equations. A photorealistic rendition of the deck surface point cloud is shown in Figure 6.6 (c).

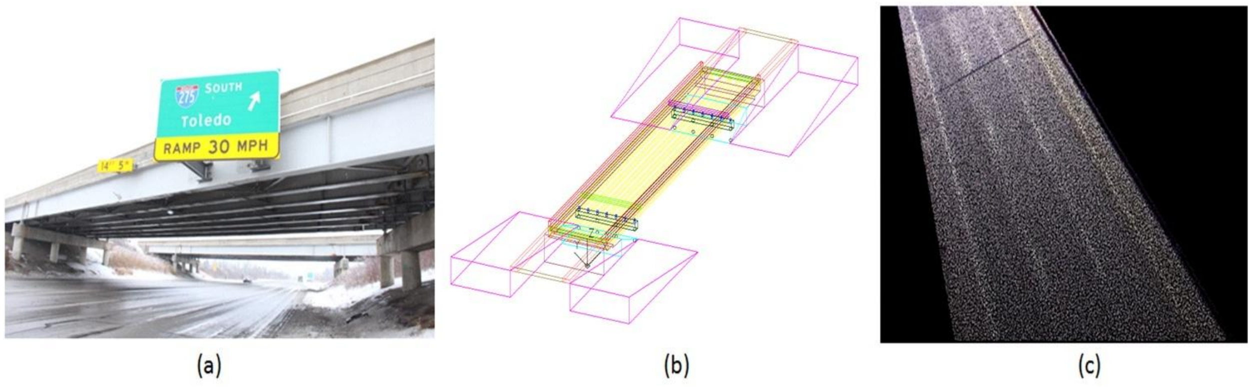


Figure 6.6: (a) The CSB, (b) the CAD model of the CSB and (c) a photorealistic rendition of the deck surface's point cloud

6.3.5 Registration to Common Reference Frame

A common problem that accompanies the use of multiple tracking and geometric data capture technologies is the necessity to register them to a common frame of reference. In this research, the author designated the native orthogonal linear coordinate frame of the CAD model as the common frame. UWB-based localization and magnetic orientation tracking technologies are orthogonal linear coordinate frame systems, and registering them to the common frame is a standard matrix transformation described in computer graphics literature. However, the GPS coordinate system is not a linear frame of reference and its registration to the common frame cannot be accomplished through such elementary geometric transformations.

The traditional solution to the GPS registration problem is the use of Vincenty's inverse algorithm, which determines the vector between two points whose GPS coordinates are known. The vector is projected onto a localized cardinal frame of reference that is registered to the common frame through a simple linear transformation. Vincenty's inverse algorithm performs an iterative computation every time the inspector acquires new GPS coordinates and registers the inspector's location to the common frame (Vincenty 1975). In this research the author implemented an alternative algorithm, presented in Appendix C, to register GPS coordinates to the common frame (Akula et al 2013). When new GPS coordinates are acquired, this registration algorithm requires only a single transformation; this offers a contrast to the iterative methodology of Vincenty's algorithm, and results in a lower consumption of computational and battery power (Akula et al 2013). The nominal accuracy of this registration algorithm was found to be comparable to that of Vincenty's inverse algorithm (Akula et al 2013). However, a detailed discussion on the accuracy and effects of registration is beyond the scope of this chapter.

6.3.6 Identifying Contextual Components

The framework processes the point cloud to compute the list of points that are located in the inspector's spatial context, as shown in Figure 6.7. As the framework performs a point-wise containment check, large point clouds would result in a high run-time deteriorating the framework performance. In order to limit the size of the point cloud being processed, large bridges are divided into overlapping computation regions—each region defined by a bounding box volume. The bridge geometry is stored as a set of point clouds corresponding to these computation regions. The computation region is determined based on the inspector's location, and the corresponding point cloud is selected for processing.

A point P in the point cloud, shown in Figure 6.5, is considered to be of contextual relevance if the following conditions are met: 1) the distance between the inspector's eye and the point as projected onto the line of sight (IP') is greater than the near plane distance (IN) and less than the far plane distance (IF) and 2) the angle between the vector from the inspector's eye to the point (IP) and the line of sight vector (IP') is less than the angular field of view (half-angle of the cone) of the inspector. The components corresponding to the points contained within the viewing cone-frustum are qualified as contextual components.

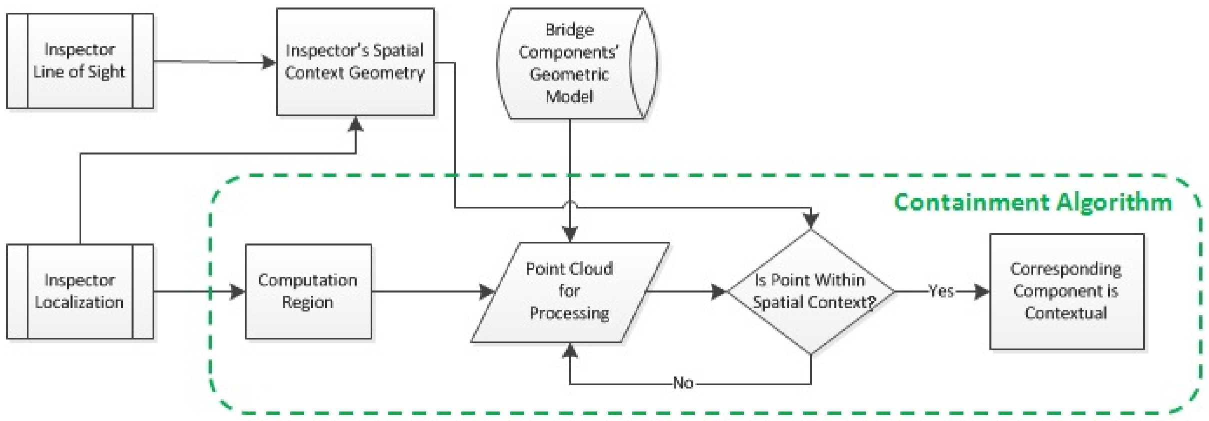

Inspector Line of Sight
Inspector's Spatial Context Geometry
Bridge Components' Geometric Model
Containment Algorithm
Inspector Localization
Computation Region
Point Cloud for Processing
Is Point Within Spatial Context?
Yes
No
Corresponding Component is Contextual

Figure 6.7: Containment algorithm to determine contextual components

6.3.7 Prioritizing Contextual Components

The framework groups the contextually relevant points by component, and then logs the following two parameters: 1) the point's distance from the line of sight (PP' in Figure 6.5), and 2) the distance between the inspector's eye and the point as projected onto the line of sight (IP' in Figure 6.5). The framework then prioritizes the contextual components in three steps.

1. The highest priority is assigned to the component that contains the point closest to the line of sight. For example, say P_1 and P_2 are contextual points—closest to the line of sight—among all points belonging to point clouds representing components C_1 and C_2, respectively. Then, C_1 is given a higher priority than C_2 if P_1 is closer to the line of sight than P_2.

2. Components whose closest points to the line of sight are within the point cloud separation distance are given the same priority to eliminate the effects of poor point cloud resolution. If the difference between P_1 and P_2's distance to the line of sight is considered statistically insignificant (less than the point cloud's resolution), then their corresponding components C_1 and C_2 are assigned the same priority.
3. In case of a tie in step 2, the prioritization algorithm assigns higher priority to the point with the minimum distance between the inspector's eye and the point as projected onto the line of sight. If components C_1 and C_2 are assigned the same priority in step 2, and if the distance between the inspector's eye and P_1 as projected onto the line of sight is less than that of P_2, then C_1 is assigned a higher priority than C_2.

6.4 Evaluation of the Context-Aware Computing Framework

6.4.1 Run-time and Space Complexity

The framework's run-time and space complexity is based on the performance of the containment and prioritization algorithms. As mentioned previously, the run-time of the containment algorithm is dependent on the size of the point cloud being processed—which in turn depends on the number of components in the computation region, the size of these components, and the point cloud resolution.

The CSB is divided into two computation regions: 1) GPS-available and 2) GPS-denied. The GPS-available region contains the point cloud data for components belonging to deck and approach components groups, whereas the GPS-denied region contains the point cloud data for components belonging to superstructure and substructure component groups. The run-time performance of the containment algorithm for GPS-available and GPS-denied regions is determined for different point cloud resolutions, and is shown in Figures 6.8 and 6.9, respectively.

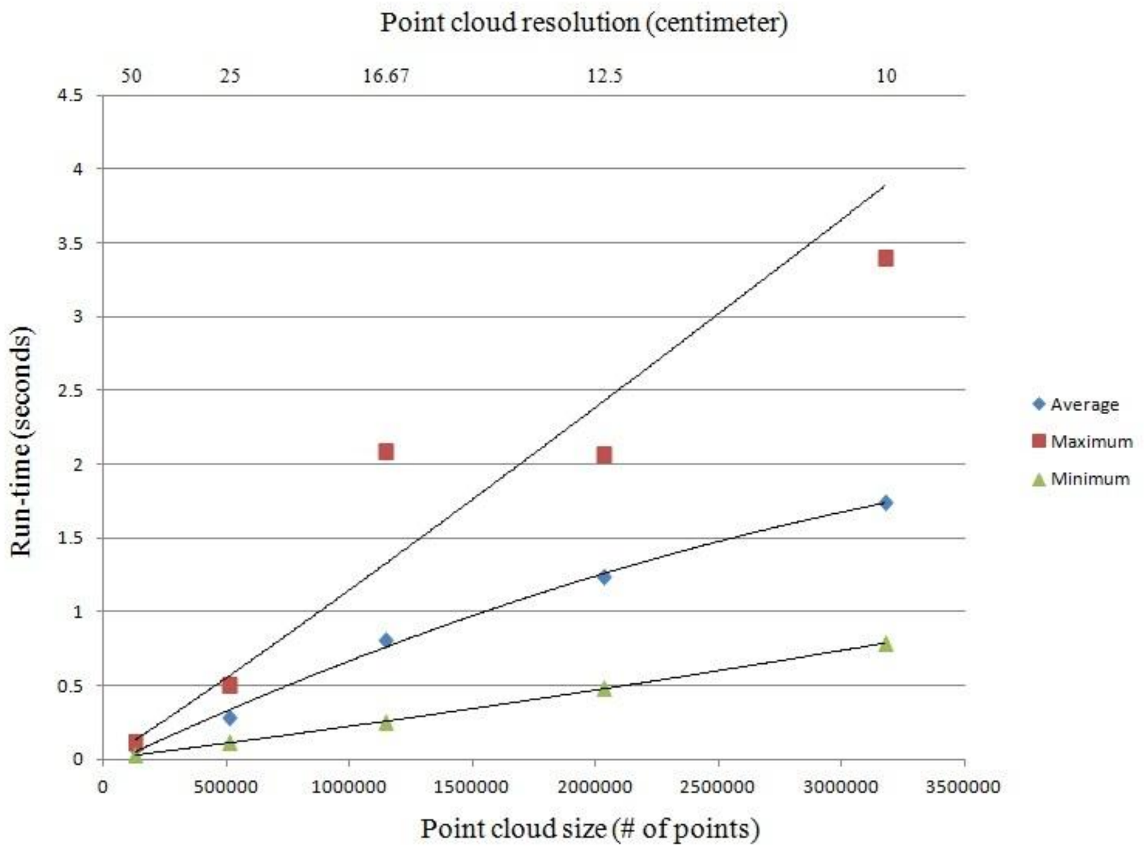


Figure 6.8: Run-time performance of the containment algorithm for GPS-available computation region

For the remainder of this chapter, the research was conducted using the point cloud with a point spacing of 20 cm, as that would allow the author to capture the smallest bridge component while still giving a sub-second–level run-time performance. The average run-time of the containment algorithm for the selected point cloud is around 0.5 seconds on a regular tablet-PC with a 1.20 GHz processor and 4.00 GB RAM, making the containment algorithm a near real time process.

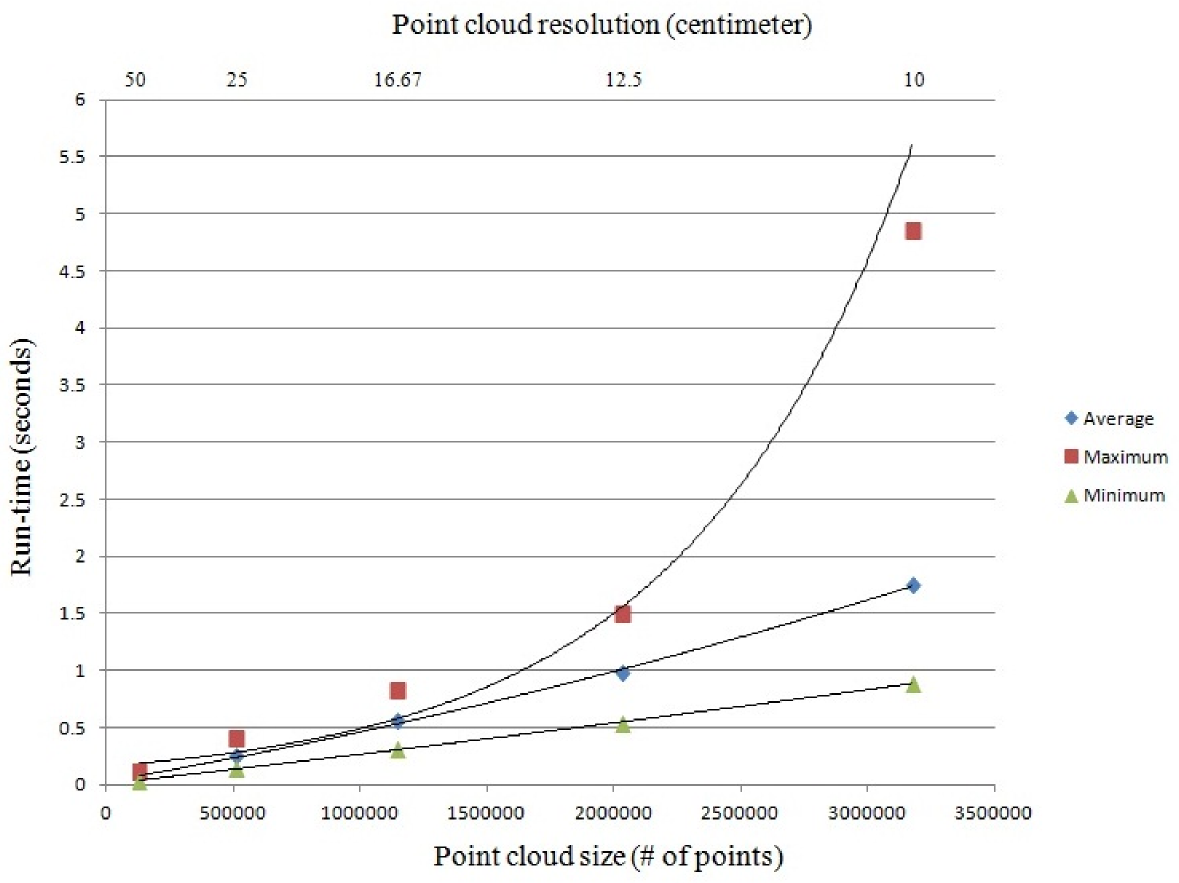


Figure 6.9: Run-time performance of the containment algorithm for GPS-denied computation region

The run-time performance of the prioritization algorithm depends on the number of components contained in the inspector’s spatial context. The average run-time of the algorithm is determined for a varying number of contextual components and is shown in Figure 6.10. The run-time of the prioritization algorithm is in the sub-millisecond range and is negligible compared to the run-time of the containment algorithm.

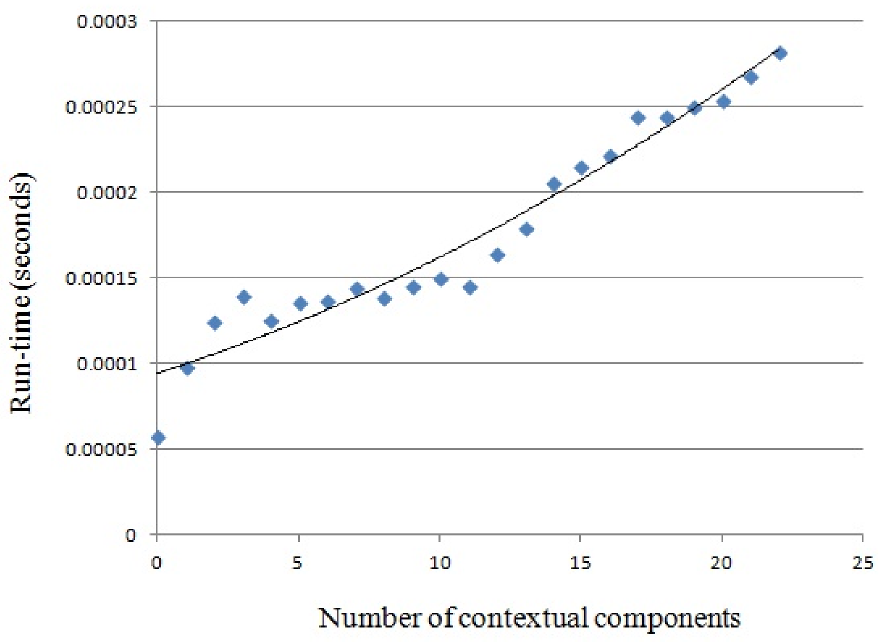


Figure 6.10: Average run-time performance of the prioritization algorithm

6.4.2 Evaluation of the Framework's Prediction

The ability of the framework to accurately predict the component of interest to the inspector depends on the precision with which the spatial context can be constructed, which in turn depends on the accuracy of the tracking technologies employed. For example, consider the construction of the spatial context of a mobile user, which is the user's viewing region, as shown in Figure 6.11 (a). Error in tracking the location and line of sight induces error in the constructed spatial context, as shown in Figures 6.11 (b) and 6.11 (c), respectively. A combination of these errors may add or cancel each other's effects while contributing to an error in the interpreted spatial context, as shown in Figures 6.11 (d) and 6.11 (e), respectively.

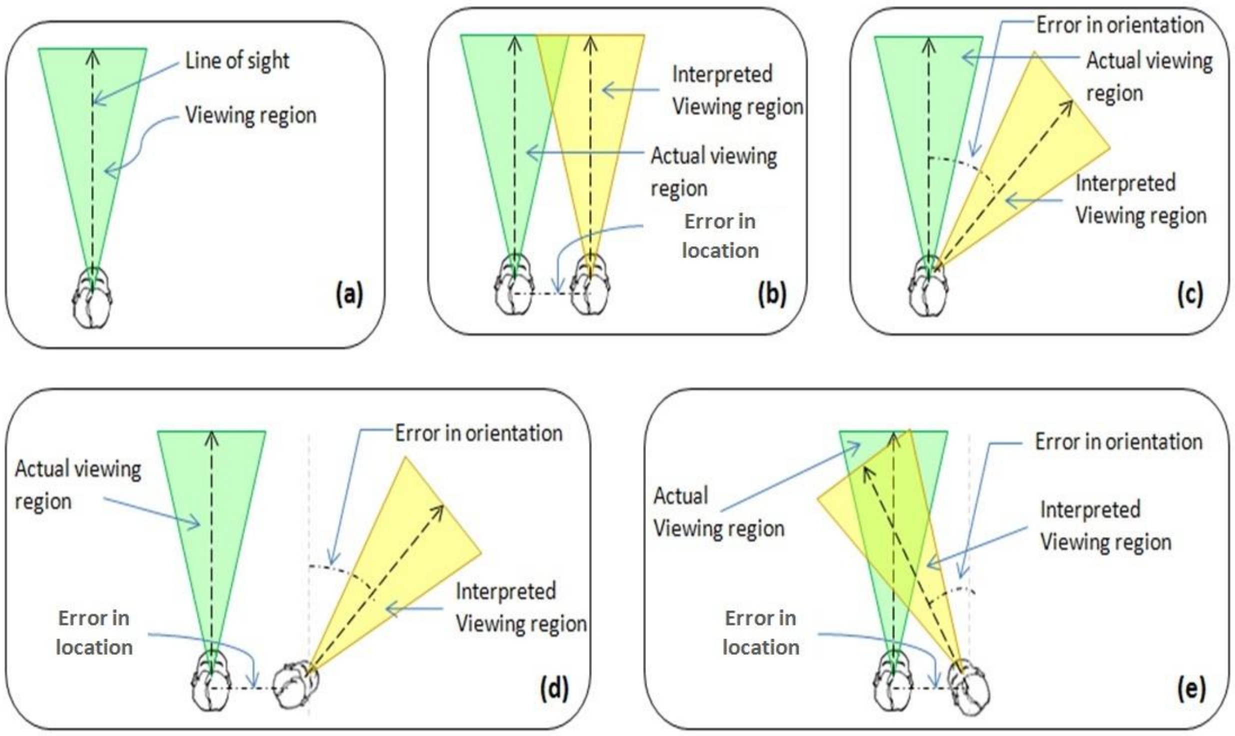


Figure 6.11: Errors in tracking result in erroneous interpretation of spatial context

While there is plenty of literature that characterizes the nature of errors inherent in the tracking technologies (Khoury and Kamat 2009b; Mautz 2009), it is difficult to predict the effects of error in tracking the pose on the interpretation of spatial context. In order to characterize the prediction performance, the author used the CSB to conduct sensitivity analysis in two stages: 1) simulation validation study and 2) field validation experiment.

6.4.2.1 Establishing Ground Truth

The sensitivity analysis is conducted for 508 inspector poses where the inspector assesses bridge components during a typical inspection routine of the CSB. The 508 poses are documented and marked on the CAD model of the CSB. Among the 508 poses, 288 locations are in GPS-available regions whereas the remaining 220 locations are in GPS-denied regions. The component of interest for the ground truth poses was determined by plotting the line of sight in the CAD model and physically verifying the inspector's object of interest.

6.4.2.2 Sensitivity Analysis: Simulation Validation Study

A simulation study was conducted to characterize the effect of uncertainties in tracking systems employed on the accuracy of predicting the component of interest. The study simulates the uncertainties in tracking technologies by sampling tracking errors from the error distribution models corresponding to the technologies employed (Akula et al 2013).

The distribution of GPS errors is modeled as an approximation of a bi-variate normal distribution with no correlation between the two variables. With this approximation, the error distribution is modeled as a Weibull distribution with a shape factor of β = 2, otherwise called the Rayleigh distribution, described by Eq. (6.1), and plotted in Figure 6.12 (Wilson 2012). The manufacturer-specified RMS error for RTK-GPS and WAAS-capable GPS is 5 cm and 3 m respectively (Akula et al 2013; Mehaffey et al 2012).

$$Probability\ (Error\ \leq\ Distance) = 1 -\ e^{-(\frac{Distance}{RMS_{Error}})^2} \qquad (6.1)$$

The UWB-based positioning system was modeled as having a uniform error distribution of 50 cm, which is reported as the maximum error in positioning using the system (Khoury and Kamat 2009b). The angular error in the electro-magnetic tracker was also sampled as a Rayleigh distribution with a manufacturer-specified angular RMS error of 0.3° (PNI Sensor Corporation

2012). The directions of the error in all of the aforementioned models were sampled randomly from a uniform distribution over 360° (Akula et al 2013).

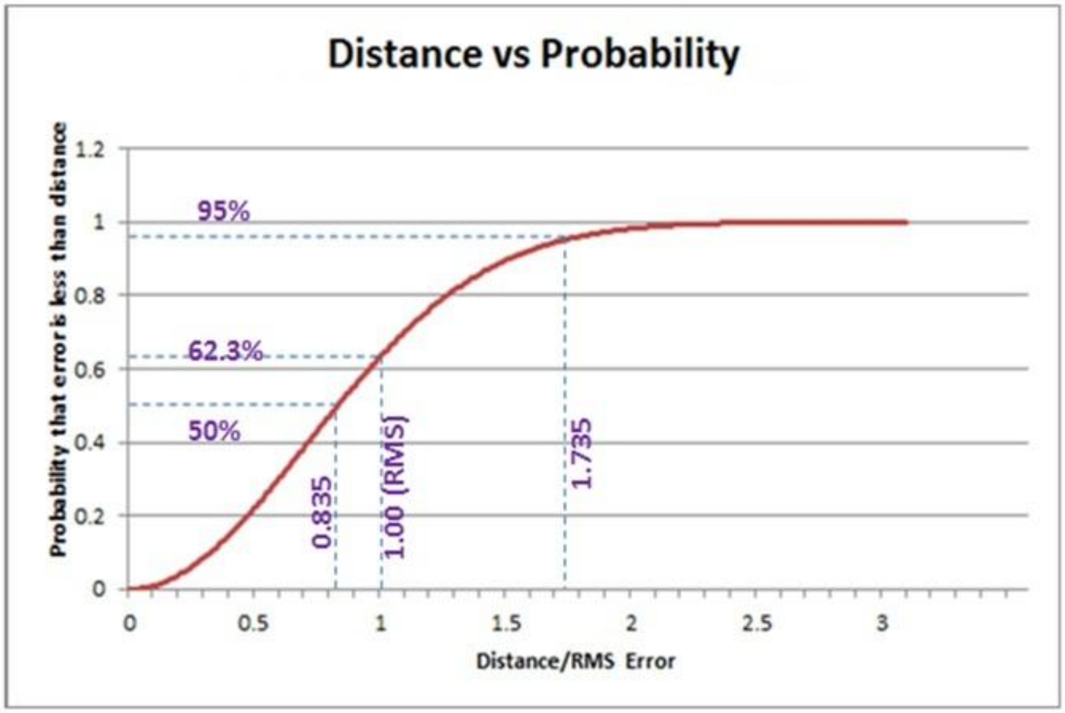


Figure 6.12: The Rayleigh distribution used to model horizontal errors in GPS

For each of the 508 poses, the simulation samples tracking errors from the corresponding error distribution models and adds them to the location and line of sight of the pose to determine the simulated pose. The simulated pose is used as the input to the framework to determine the predicted component of interest. The simulation is repeated 100 times for all 508 poses and the framework's prediction is compared to the ground truth. In order to determine the optimum view angle parameter of the framework, the simulation validation study performance is determined for view angles of 30^o, 35^o, 40^o, 45^o, 50^o, 55^o, and 60^o (Akula et al 2013).

6.4.2.3 Sensitivity Analysis: Field Validation Experiment

Using the CSB as a test bed, the author conducted a validation experiment to characterize the effects of uncertainty in tracking on the accuracy of the framework's performance (in predicting the component of interest). As mentioned previously, GPS coverage was available for 288 of 508 ground truth poses. The field experiment was conducted for these 288 points using GPS-based position tracking systems. The field experiment could not be conducted for the remaining 220 poses in GPS-denied environments due to limited access and MDOT's permission restrictions on installing infrastructure for a UWB-based localization system.

The mobile inspector is fitted with a body suit that ergonomically holds the tracking system hardware, as shown in Figure 6.3. The GPS unit is fixed on top of the inspector's helmet and is connected to the computation center, which is a tablet-PC. The electro-magnetic compass is enclosed in an aluminum container and is placed at the lowest point on the inspector's helmet, directly above the forehead. This ensures that the orientation of the tracker is as close to the user's line of sight as possible.

For the 288 poses where GPS is accessible, the ground truth poses are physically marked on the CSB. The inspector assumes the marked pose and communicates to the framework to lock the pose parameters. The framework acquires the inspector's location and line of sight as determined by the tracking systems, and then predicts the component of interest. This process is repeated for all 288 poses and the framework's prediction is compared to the ground truth. In order to determine the optimum view angle parameter of the framework, the field experiment performance is determined for view angles of 30^o, 35^o, 40^o, 45^o, 50^o, 55^o, and 60^o.

6.5 Results of Sensitivity Analyses

6.5.1 Results of the Simulation Validation Study

The author compares the framework's prediction of the component of interest with the ground truth component of interest. The results of this comparison are classified into three categories:

1. Correct – The predicted component of interest is the same as the ground truth component of interest.
2. Contextual – The ground truth component of interest was identified to be within the spatial context of the inspector. All correct predictions are also classified as contextual.
3. Non-contextual – The ground truth component of interest was not identified to be within the spatial context of the inspector.

Figure 6.13 summarizes the results of the framework's prediction performance in the GPS-available computation region when the analysis models the location as being tracked by RTK-GPS. The framework correctly predicts the component of interest in 98% of the cases and always classifies it as contextual (Akula et al 2013).

Figure 6.14 summarizes the results of the framework's prediction performance in the GPS-available computation region when the location is modeled as being tracked by WAAS GPS. The framework correctly predicts the component of interest in 72% of the cases and classifies it as contextual in about 98% of the cases (Akula et al 2013).

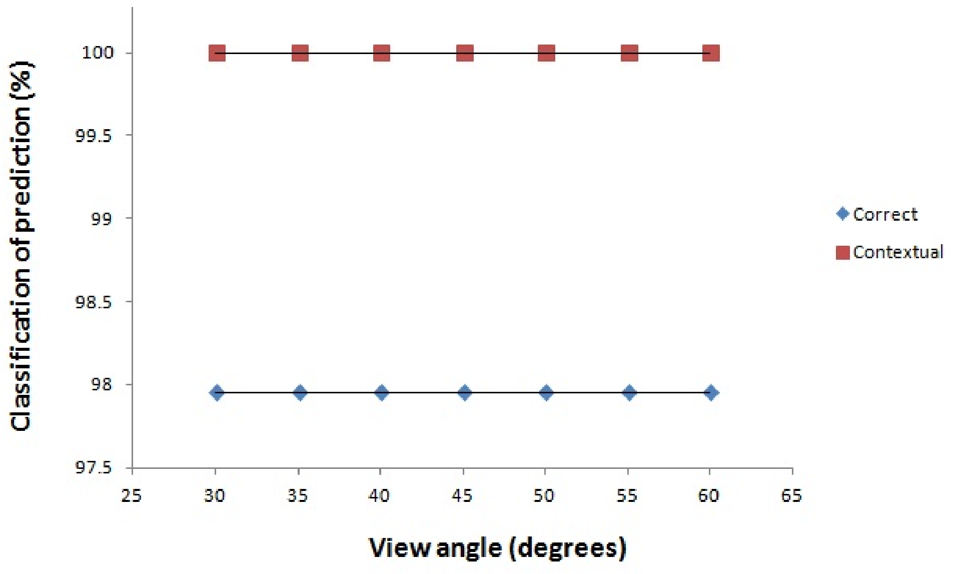


Figure 6.13: Framework's simulated prediction performance for RTK-GPS localization in GPS-available region (Akula et al 2013)

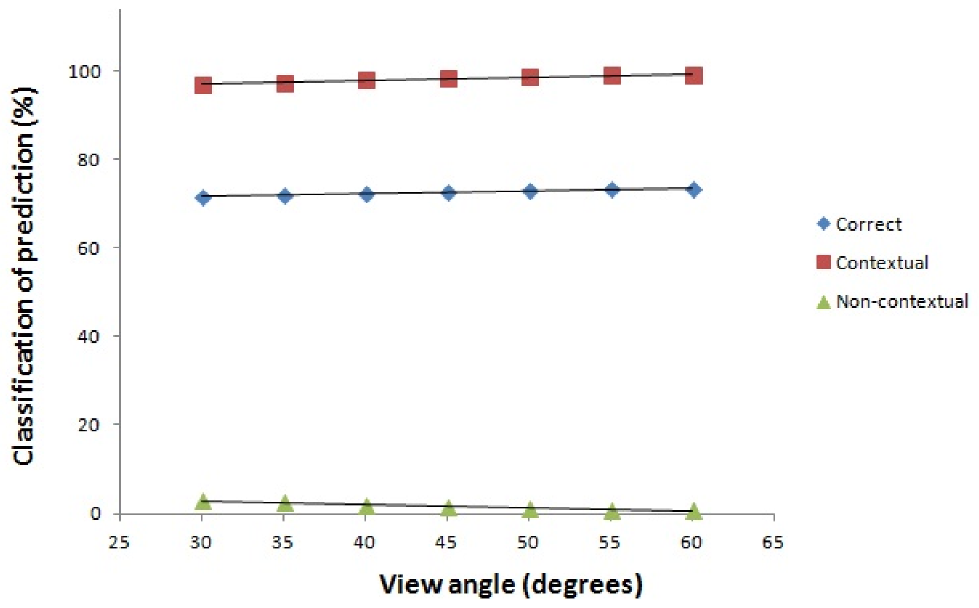


Figure 6.14: Framework's simulated prediction performance for WAAS GPS localization in GPS-available region (Akula et al 2013)

Figure 6.15 summarizes the results of the framework’s prediction performance in the GPS-denied computation region when the location is modeled as being tracked by a UWB localization system. The framework correctly predicts the component of interest in 90% of the cases and classifies it as contextual in about 99.5% of the cases (Akula et al 2013).

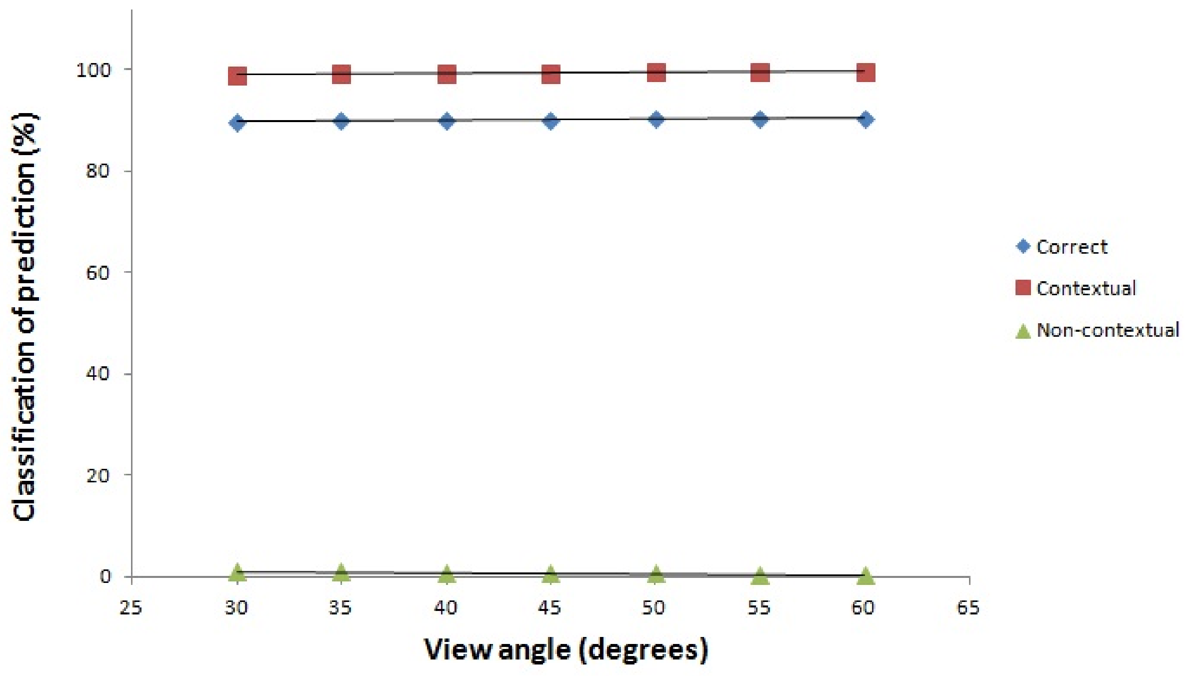


Figure 6.15: Framework’s simulated prediction performance for UWB localization in GPS-denied region (Akula et al 2013)

6.5.2 Results of the Field Validation Experiments

Figure 6.16 summarizes the results of the framework’s prediction performance in the field experiments when the location and line of sight were tracked by RTK-GPS and the magnetic compass, respectively. The framework correctly predicts the component of interest in 56% of the cases and classifies it as contextual in 70% to 80% of the cases based on the view angle parameter. The prediction accuracy of the framework was found to be 40% poorer than the accuracy predicted by the simulation study for the same tracking technologies.

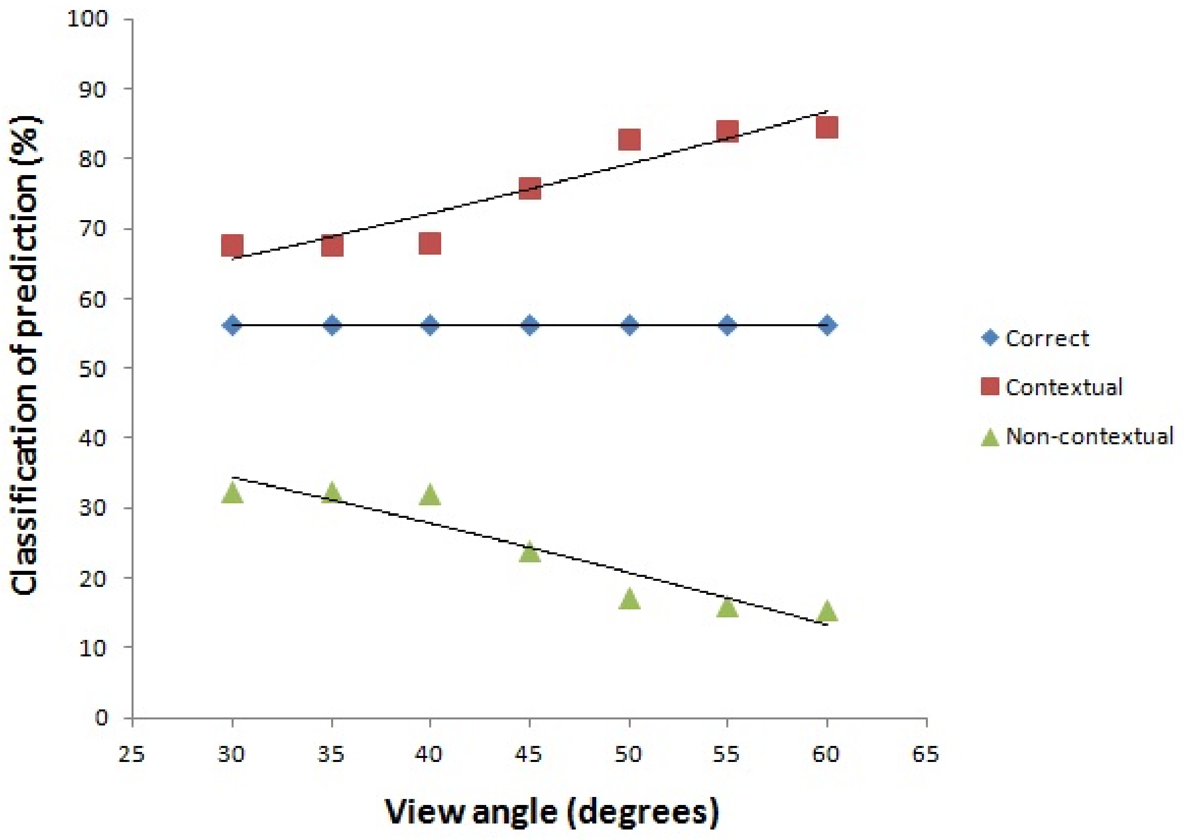


Figure 6.16: Framework's prediction performance in the field when localization and line of sight are tracked using RTK-GPS and magnetic compass

Figure 6.17 summarizes the results of the framework's prediction performance in the field experiments when the location and line of sight were tracked by WAAS GPS and the magnetic compass, respectively. The framework correctly predicts the component of interest in 40% of the cases and classifies it as contextual in 70% to 85% of the cases based on the view angle parameter. The prediction accuracy of the framework was found to be 30% poorer than the accuracy predicted by the simulation study for the same tracking technologies.

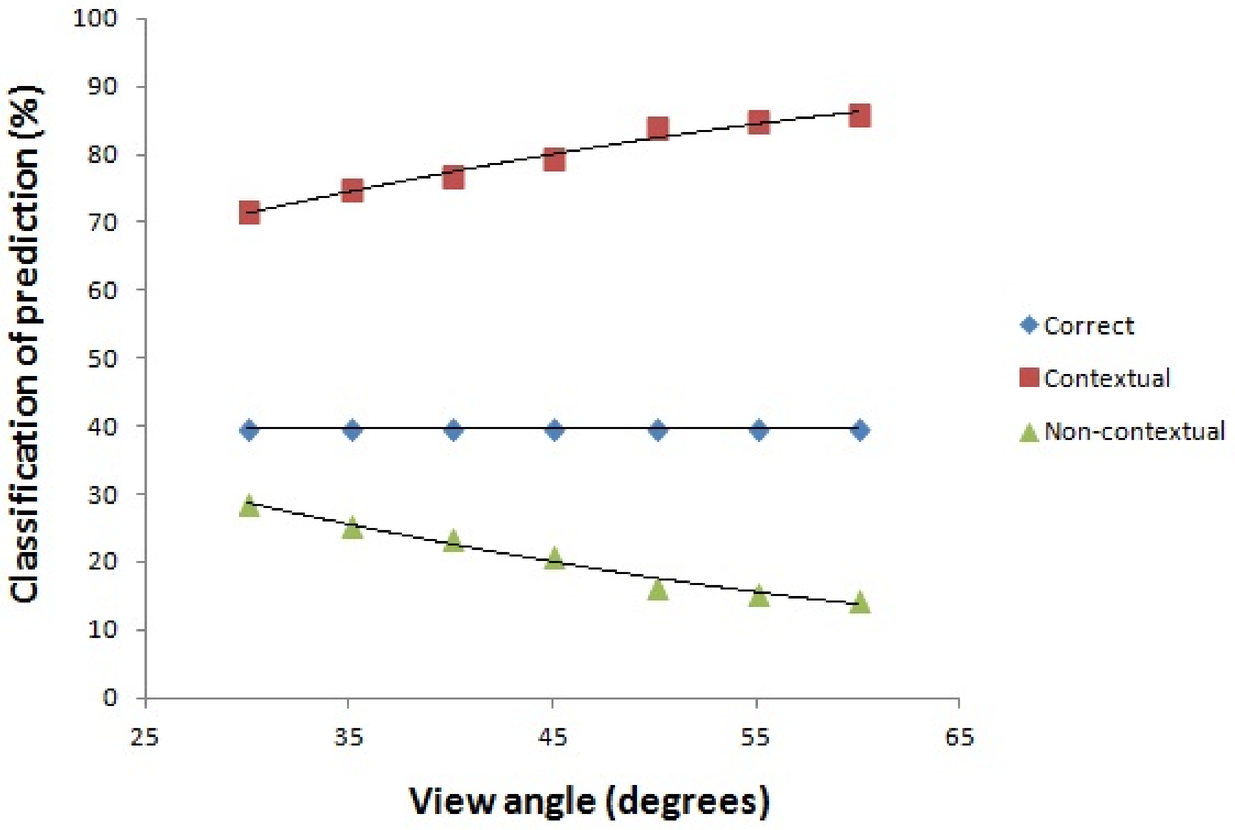


Figure 6.17: Framework's prediction performance in the field when localization and line of sight are tracked using WAAS GPS and magnetic compass

In order to better understand the discrepancy between the framework's performance as predicted by the simulation study and the performance in the field, the author decoupled the location tracking technology from the line of sight tracking technology. The author uses field readings for localization while sampling line of sight readings through the simulation model (Akula et al 2013). The resulting field location and simulated line of sight readings are used as inputs to the framework to predict the component of interest. The framework's predictions are compared to ground truth component of interest. Figure 6.18 summarizes the results of the framework's prediction performance when the location was tracked by RTK-GPS and line of sight was sampled using the simulation model. The framework correctly predicts the component of interest in 97% of the cases and classifies it as contextual in nearly all of the cases (Akula et al 2013).

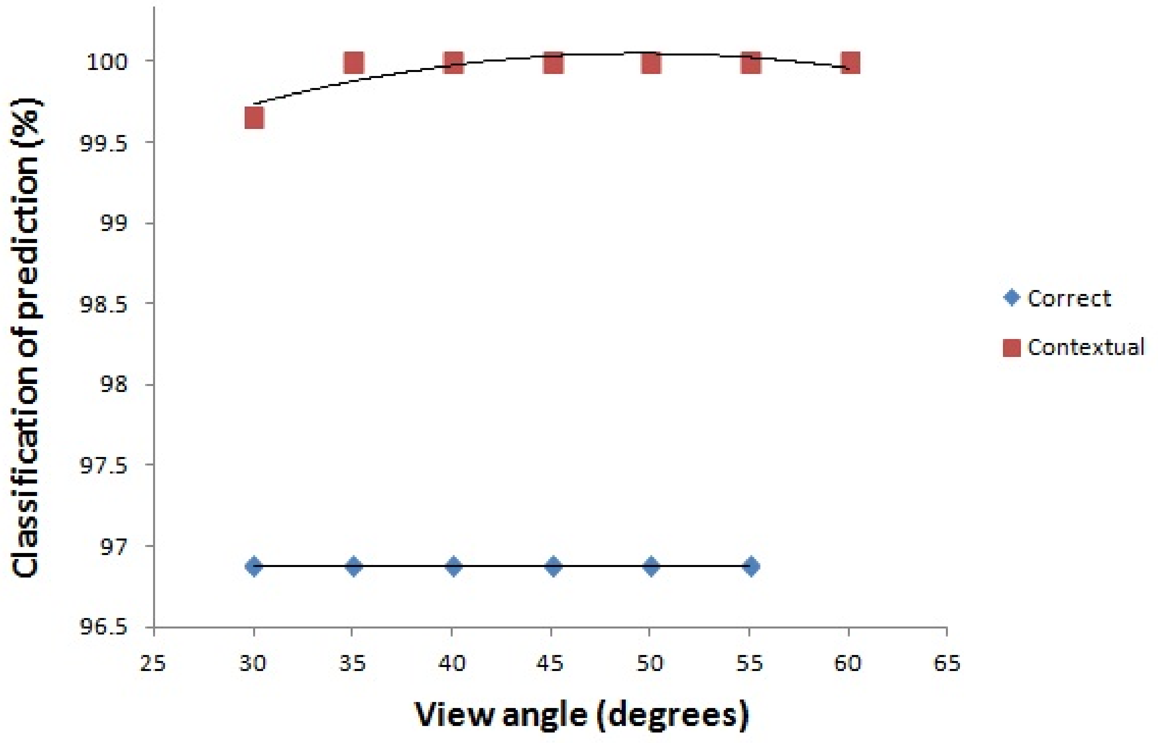


Figure 6.18: Framework's prediction performance when location is tracked using RTK-GPS and line of sight is sampled using the simulation model (Akula et al 2013)

Figure 6.19 summarizes the results of the framework's prediction performance when the location was tracked by WAAS GPS and line of sight was sampled using the simulation model. The framework correctly predicts the component of interest in 65% of the cases and classifies it as contextual in 98% of the cases (Akula et al 2013).

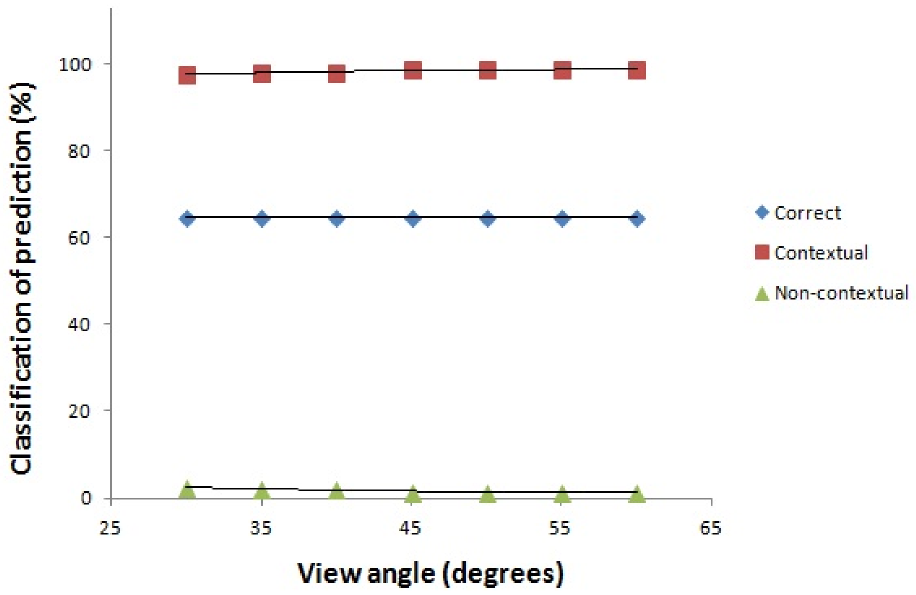


Figure 6.19: Framework's prediction performance when location is tracked using RTK-GPS and line of sight is sampled using the simulation model (Akula et al 2013)

The framework's performance in these cases is comparable to the performance predicted by the simulated validation study. This confirms that the deterioration of the framework's performance during the field validation experiments is mainly due to the erratic behavior of the magnetic compass used to track the inspector's line of sight in the field (Akula et al 2013). The average nominal error of the magnetic compass in the field was found to be around 2°—about 6 times the manufacturer-specified RMS error used for the simulated validation study. The magnetic compass relies on the earth's magnetic field to calculate the orientation readings and thus becomes unreliable in the field as the magnetic fields of the bridge deck rebar, structural steel, and traffic interfere with the earth's magnetic field (Akula et al 2013).

6.5.3 Limitations of the Results

Framework design: The tracking technologies employed by the framework result in an increased payload carried by the mobile inspector, which may result in limiting the ability to perform certain physical movements and tasks. The framework approximates the complex phenomenon of human vision by using a truncated cone, considered representative of the inspector's viewing region. The study assumes that upon approaching a component of interest, the inspector can be trained to align the head orientation with the line of sight before communicating to the framework to lock the pose.

Data homogeneity: The research is limited by the use of as-designed CAD models instead of the as-built CAD models to develop the point cloud of the components. The ground truth components of interest were determined using as-designed CAD models. However, the field experiments were conducted by marking these points, relative to bridge features, on the built structure. The effects of the discrepancies between as-built and as-designed data on the framework's performance are unknown.

Registration methodology: The framework utilizes a common coordinate frame to which the point cloud mapping and the tracking technologies are registered. This process introduces registration errors into pose tracking data and it is unknown how the registration process affects the framework's performance. Methods of quantifying GPS registration errors (i.e., how registration errors propagate into measured data) do not currently exist.

Simulation model: The simulation validation study uses idealistic theoretical distributions based on manufacturers' specifications to characterize the errors in the tracking technologies. However, as demonstrated by the field experiment, this is not always true. The magnetic compass used to determine the head orientation behaves erratically when exposed to field conditions due to magnetic influences from rebar, structural system, and vehicle traffic.

Field validation: As mentioned previously, the field experiment results are limited by the fact that field data was unavailable in GPS-denied environments due to limited access and permission restrictions from MDOT for installing UWB-based localization system infrastructure on the CSB. While the simulation validation study was conducted by repeating error sampling 100 times for each pose, the field experiment data was sampled only once, and the results do not take repeatability into account.

Case study environment: Finally, the results of the research presented in this chapter are also limited by the fact that all simulations and experiments were conducted on the CSB. Conducting the simulation test, presented in this chapter, for multiple bridges was logistically difficult as it would have required the point clouds (from said bridges) to be captured and mapped to the corresponding components. Conducting field tests for multiple bridges would result in the additional difficulty of requiring MDOT staff to accompany the author on the bridge for traffic control and safety reasons when the author's conduct imitated bridge inspection routines. However, as mentioned previously, the CSB was recommended to the author by MDOT as the best topological representative of most highway bridges in the state of Michigan. The error in predicting the object of interest depends on the topology of the components. Therefore, the author considers the range of the results reported in this research to be applicable for most regular Michigan highway bridges because they are topologically similar to the CSB. Methods to quantify the relationship between component topology and the accuracy of predicting the component of interest do not currently exist.

6.6 Summary and Conclusions

Context-aware computing has the potential to improve the efficiency of complex and tedious processes by providing streamlined information relevant to the task at hand. Spatial context is one of the most important parameters considered in context-aware computing, and is used by such applications to determine the entities of interest in order to retrieve and present the user with appropriate data relevant to the task at hand. The author presents a motivating scenario of conducting bridge inspection routines where integrating context-aware computing with new and existing bridge inspection reporting applications can potentially save the time spent in manually navigating the software, and can thus reduce the effort spent by inspectors to learn, remember, and recall the taxonomic hierarchy and ontology of bridge components across a large fleet of bridges.

The author presents the architecture of a context-aware computing framework to automatically identify bridge components in the inspector's spatial context. As part of the framework, the author presents methods to map bridge component geometry and to register various point cloud mapping and contextual parameter tracking technologies to a common coordinate frame. The author then presents the containment and prioritization algorithms used by the framework to predict the component of interest to the inspector. The chapter then presents run-time and spatial complexity analysis of the framework.

The chapter also covers a sensitivity analysis study that characterized the framework's performance in predicting the component of interest. The sensitivity analysis is performed using two approaches: the simulation study approach and the field experiment approach. The aforementioned approaches document the predictions of the framework by sampling errors in the tracking technologies through simulation and field recordings respectively. The framework's predictions during the simulation study and field experiment are then compared to the ground truth components of interest. The author then presents a discussion on the limitations of the results presented in the sensitivity analysis. The effect of using RTK-GPS, WAAS GPS, and magnetic compass tracking technology on the framework's prediction performance is summarized in Table 6.3. Based on the research presented in this chapter, the author concludes that the use of a magnetic compass to track the line of sight is unfeasible, and suggests exploring the use of high-accuracy gyroscopes, accelerometers, and eye-ball tracking technologies.

	RTK-GPS	WAAS GPS	Magnetic compass
Payload addition	Approx. 1.5 kg	Approx. 100 grams	Less than 100 grams
Infrastructure required	RTK base station coverage	WAAS reference stations are maintained by the US government	N/A
Manufacturer specified error	RMS 5 cm	RMS 3 m	RMS 0.3°
Error characteristics in field	Close to manufacturer-specified error	Close to manufacturer-specified error	Nominal error was around 2°
Effect on framework prediction performance	Deteriorates by 1% to 2%	Deteriorates by less than 30%	Deteriorates by 30% to 40%

Table 6.3: The summary of GPS and magnetic compass tracking technologies

The author suggests integrating the framework with bridge inspection reporting applications, as illustrated in Figure 6.20. The framework is integrated such that when the inspector locks the pose, the framework predicts the component of interest and automatically queries the inspection database with the component of interest. The queried data is retrieved and displayed to the inspector. If the predicted component of interest is not the inspector's component of interest, the inspector can choose to retrieve the list of components classified as contextual. The inspector can then select the component of interest and retrieve the appropriate information. If the component of interest is not classified as contextual, then inspector can retrieve the list of components in the computation region and select the component of interest.

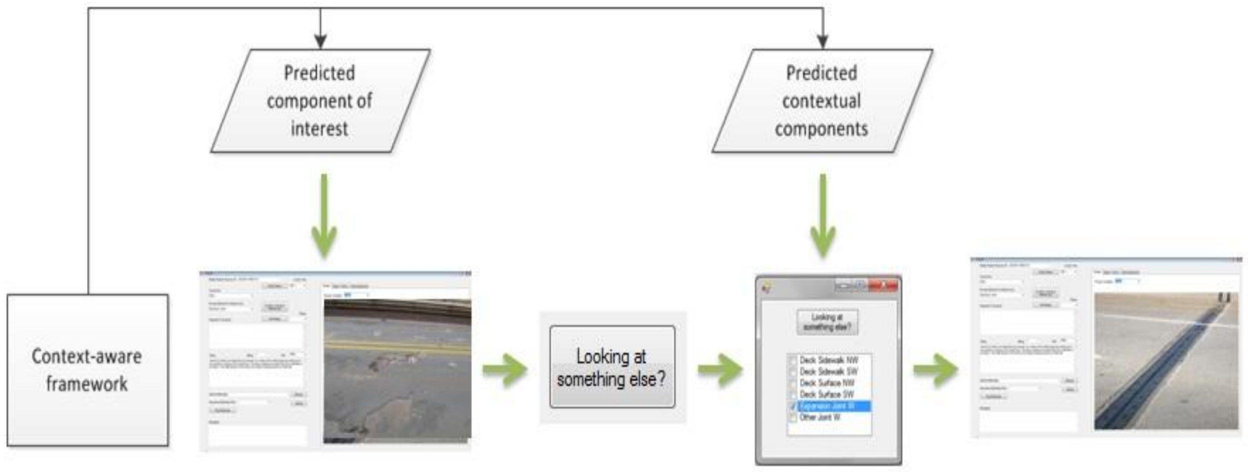


Figure 6.20: Integrating context-aware computing framework workflow

Supported by high-precision tracking technologies, the framework would automatically identify the component of interest in all but the rarest of cases, thus reducing the time required to navigate the software and the effort spent by inspectors to learn, remember, and recall the taxonomic hierarchy and ontology of bridge components. The development of the context-aware computing framework and the sensitivity analysis presented in this chapter accomplishes an important exploratory step in the design and integration of context-aware computing in bridge inspection reporting and other civil engineering domain applications.

6.7 Acknowledgments

The author would like to gratefully acknowledge the support offered by the U.S. Department of Commerce, National Institute of Standards and Technology (NIST) Technology Innovation Program (TIP) under Cooperative Agreement Number 70NANB9H9008. Additional support was provided by the University of Michigan and the Michigan Department of Transportation (MDOT). The author would also like to thank Atul Sandur and Prof. Atul Prakash for their valuable contribution to conducting the sensitivity analysis study. The intellectual contributions offered by Dr. Sharada Alampalli and Dr. Mohammed Ettouney are greatly appreciated. Any opinions, findings, conclusions, and recommendations expressed in this chapter are those of the author and do not necessarily reflect the views of the institutions or the individuals mentioned here.

6.8 References

AASHTOWare (2012). "AASHTOWare™ Bridge Management Software" – Website of the AASHTOWare™ bridge management software, <http://pontis.inspecttech.com/> as accessed on 10th September, 2012.

Adams, T.M., Juni, E., Siddiqui, M.K., and Dzienkowski, J.E. (2005). "Integrated Field and Office Tools for Bridge Management," Transportation Research Record, Transportation Research Board, Vol. 1933(1), pp. 35–43.

Akula, M., Dong, S., Kamat, V.R., Ojeda, L., Borrell, A. and Borenstein, J. (2011). "Integration of Infrastructure based Positioning Systems and Inertial Navigation for Ubiquitous Context-Aware Engineering Applications," Advanced Engineering Informatics, Elsevier, Vol. 25(4), pp. 640–655.

Akula, M., Sandur, A., Kamat, V.R., and Prakash, A. (2013). "Context-Aware Framework for Highway Bridge Inspections," Journal of Computing in Civil Engineering, American Society of Civil Engineers (ASCE), Reston, VA (In Press).

Aittola, M., Ryhanen, T., and Ojala, T. (2003). "SmartLibrary – Location-Aware Mobile Library Service," Human-Computer Interaction with Mobile Devices and Services, Springer, Berlin, Germany, pp. 411–416.

Atasoy, G., Tang, P., Zhang, J., and Akinci, B. (2010). "Visualizing Laser Scanner Data for Bridge Inspection," The 27th Int. Symp. on Automation and Robotics in Construction (ISARC 2010), Bratislava, Slovakia.

Aziz, Z., Anumba, C.J., Ruikar, D., Carrillo, P.M., and Bouchlaghem, D.N. (2005). "Semantic Web-based Services for Intelligent Mobile Construction Collaboration," Information Technology in Construction (ITCon): Special Issue on Mobile Computing in Construction, Vol. 9, pp. 367–379.

Aziz, Z., Anumba, C.J., Ruikar, D., Carrillo, P., and Bouchlaghem, D.N. (2006). "Intelligent Wireless Web Services for Construction – A Review of Enabling Technologies," Automation in Construction, Elsevier, New York, NY, Vol. 15(2), pp. 113–123.

Bachmann, T., Naab, K., Reichart, G., and Schraut, M. (2000). "Enhancing Traffic Safety with BMW's Driver Assistance Approach Connected Drive," Proc. of the 7th World Congress on Intelligent Transportation Systems Conference, Turin, Italy, pp. 21–24.

Bardram, J.E., and Christensen, H.B. (2007). "Pervasive Computing Support for Hospitals: An Overview of the Activity-based Computing Project," IEEE Pervasive Computing, Institute of Electrical and Electronics Engineers (IEEE), Vol. 6(1), pp. 44–51.

BridgeInspect^{TM} (2012). "BridgeInspect Software for Bridge Inspectors and Managers" – Website of BridgeInspect^{TM} <http://www.bridgeinspect.com/> as accessed on 15th December, 2012.

BridgeWorks.NET (2012). "Washington State Department of Transportation's Bridge Inspection and Reporting Software" – Website of the Washington State Department of Transportation <http://www.wsdot.wa.gov/LocalPrograms/Bridge/BridgeWorks.htm> as accessed on 15th December, 2012.

Burrell, J. and Gay, K. (2001). “Collectively Defining Context in a Mobile, Networked Computing Environment,” Proc. of the Conference on Human Factors in Computing Systems, Association for Computing Machinery (ACM), New York, NY, pp. 231–232.

Davies, N., Cheverst, K., Mitchell, K., and Friday, A. (1999). “Caches in the Air: Disseminating Information in the Guide System,” Proc. of the Second IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), Institute of Electrical and Electronics Engineers (IEEE), New Orleans, LA, pp. 11–19.

Farrar, M. M. (2008). “Inspection,” The AASHTO Manual for Bridge Evaluation, 2nd Edition, with 2011 Interim Revisions, Section 4, American Association of State Highway and Transportation Officials, Washington, D.C., ISBN: 1-56051-496-1.

Fathi, M.S., Anumba, C.J., and Carrillo, P. (2006). “Context Awareness in Construction Management – Key Issues and Enabling Technologies,” Proc. of the Joint Int. Conference on Construction Culture, Innovation and Management (CCIM), Dubai, UAE, pp. 425–435.

Fleck, M., Frid, M., Kindberg, T., O’Brien-Strain, E., Rajani, R., and Spasojevic, M. (2002). “From Informing to Remembering: Deploying Ubiquitous Systems in Interactive Museums” IEEE Pervasive Computing, Institute of Electrical and Electronics Engineers (IEEE), Vol. 1(2), pp. 13–21.

Grabe, V. (2010). “Wide Field of View Head Mounted Display: Integration and Evaluation in a Motion Simulator,” Diploma Thesis in Bioinformatics, University of Tubingen, Tubingen, Baden-Wurttemberg, Germany, pp. 1–53.

Hoch, S., Schweigert, M., Althoff, F. and Rigoll, G. (2007). “The BMW SURF Project: A Contribution to the Research on Cognitive Vehicles,” Proc. of the IEEE Vehicles Symposium, Institute of Electrical and Electronics Engineers (IEEE), Istanbul, Turkey, pp. 692–697.

Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J. (1993). “Global Positioning system: Theory and Practice,” Global Positioning system: Theory and Practice, Springer, New York, ISBN: 0-387-82477-4, pp. 1–347.

Indiana State BIAS (2012). “Indiana Bridge Inspection Application System” – Website of the Indiana State Bridge Inspection Application System <https://inbridges.indot.in.gov/> as accessed on 15th December, 2012.

Jahnke, J.H., Bychkov, Y., Dahlem, D., and Kawasme, L. (2005). “Context-Aware Information Delivery in Health Care,” Revue d’Intelligence Artificielle, Vol. 19(3), pp. 459–478.

Khoury, H.M., and Kamat V.R. (2009a), "High Precision Identification of Contextual Information in Location-Aware Engineering Applications,” Advanced Engineering Informatics, Elsevier Science, New York, NY, Vol. 23(4), pp. 483–496.

Khoury, H.M., and Kamat V.R. (2009b), "Indoor User Localization for Context-Aware Information Retrieval in Construction Projects," Automation in Construction, Elsevier Science, New York, NY, Vol. 18(4), pp. 444–457.

Koch, C., and Brilakis, I. (2011). "Pothole Detection in Asphalt Pavement Images," Advanced Engineering Informatics, Elsevier Science, New York, NY, Vol. 25(3), pp. 507–515.

Laukkanen, M., Helin, H., and Laamanen, H. (2002). "Tourists on the Move," Int. Workshop Series on Cooperative Information Agents VI, Springer, Heidelberg, Germany, pp. 36–50.

Mautz, R. (2009). "Overview of Current Indoor Positioning Systems," Geodezija Ir Kartografija/ Geodesy and Cartography, Vol. 34(2), pp. 18–22.

MDOT (2011). "Highway Bridge Report – October 1, 2011," Michigan Department of Transportation (MDOT), Lansing, MI. pp. 1–2.

Mehaffey, J., Yeazel, J., Penrod, S., and Deiss, A. (2012). "Garmin's eTrex Legend C: With Horizontal Compass and Antenna" – Joe Mehaffey, Jack Yeazel, Sam Penrod, and Allory Deiss' GPS Information Website <http://gpsinformation.us/vistacolor/etrexvistacolor.html> as accessed on 25th April, 2012.

Moore, W.H. (2011). "Table 1-28: Condition of U.S. Highway Bridges," National Transportation Statistics, Bureau of Transportation Statistics, Washington, D.C.

PNI Sensor Corporation (2012), "Legacy TCM," Website of the PNI Sensor Corporation <http://www.pnicorp.com/products/tcm-legacy> as accessed on 25th April, 2012.

Rash, C.E., Bayer, M.M., Harding, T.H., and McLean, W.E. (2009) "Visual Helmet-Mounted Displays," Helmet Mounted Displays: Sensation, Perception and Cognition Issues, Chapter 4, U.S. Army Aeromedical Research Laboratory, Fort Rucker, Alabama, pp. 109–174.

Schilit, B.N., Adams, N., and Want, R. (1994). "Context-Aware Computing Applications," Proc. of the Workshop on Mobile Computing Systems and Applications, Institute of Electrical and Electronics Engineers (IEEE), Santa Cruz, CA, pp. 85–90.

Schmidt, A. (2002). "Ubiquitous Computing – Computing in Context," Ph.D. Thesis, Lancaster University, Lancaster, England, pp. 1–294.

Schwinger, W., Grun, C., Proll, B., and Retschitzegger, W. (2009). "Context-Awareness in Mobile Tourist Guides," Handbook of Research on Mobile Multimedia, Second Edition, IGI Global, Hershey, PA, pp. 534–552.

Skibniewski, M.J., and Jang, W.S. (2006). "Ubiquitous Computing: Object Tracking and Monitoring in Construction Processes utilizing ZigBee Networks," Proc. of the 23rd Int. Symp. on Automation and Robotics in Construction (ISARC 2006), Tokyo, Japan, pp. 287–292.

Stukhart, G., and Bell, L.C. (1985). “Attributes of Materials Management Systems,” A Report to the Construction Industry Institute, Texas A&M University and Auburn University.

Tang, P., and Akinci, B. (2012). “Formalization of Workflows for Extracting Bridge Surveying Goals from Laser-Scanned Data,” Automation in Construction, Elsevier Science, New York, NY, Vol. 22(3), pp. 306–319.

Taylor, G., Uff, J. and Al-Hamadani, A. (2001). “GPS Positioning using Map-Matching Algorithms, Drive Restriction Information and Road Network Connectivity,” Proc. of the 9th Annual GIS Research UK, Glamorgan, Wales, pp. 114–119.

Teizer, J., Venugopal, M., and Walia, A. (2009). "Ultra Wideband for Automated Real-Time Three-Dimensional Location Sensing for Workforce, Equipment, and Material Positioning and Tracking,” Transportation Research Record: Journal of the Transportation Research Board, No. 2081, Washington, D.C., pp. 56–64.

The Business Roundtable (1982). “Modern Management Systems,” Construction Industry Cost Effectiveness Project Report, A-6, The Business Roundtable, November 1982.

Thomas, B., Close, B., Donoghue, J., Squires, J., De Bondi, P., Morris, M., and Piekarski, W. (2002) "ARQuake: An Outdoor/Indoor Augmented Reality First Person Application,” Personal and Ubiquitous Computing, Springer Science & Business Media, Vol. 6(2), pp. 75–86.

Vincenty, T. (1975). "Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations", Survey Review, Vol. XXIII (No. 176), pp. 88–93.

Wilson, D.L. (2012), “GPS Horizontal Position Accuracy” – GPS Accuracy Webpage < http://users.erols.com/dlwilson/gpsacc.htm> as accessed on 25th April, 2012.

Zou, W., Ye, X., and Chen, Z. (2006) “A Brief Review on Application of Mobile Computing in Construction,” The 1st Int. Multi-Symposiums on Computer and Computational Sciences (IMSCCS’06), Institute of Electrical and Electronics Engineers (IEEE), Zhejiang University, Hangzhou, China, Vol. 2, pp. 657–661.

Chapter 7

Context-Aware Bridge Inspection Routines

7.1 Introduction

7.1.1 Importance of Bridge Condition Assessment

Bridges are an essential component of a nation's civil infrastructure and transit systems, affecting passenger and commercial traffic on the regional and national scales. In 2008, the United States Department of Transportation's (U.S. DOT) Federal Highway Administration (FHWA) division estimated passenger vehicles to account for approximately 56% of highway traffic in the United States (U.S.) (FHWA 2013). Although the number of commercial vehicles is smaller than passenger vehicles, the FHWA estimated that commercial track traffic carried over $11 trillion worth of freight in 2008 within the U.S. (FHWA 2013). The FHWA also predicts that most highway routes will experience increases in both passenger and commercial traffic over the next 20 years. Therefore, problems with substandard highway bridges not only affect millions of passengers in terms of safety and convenience, but can cause substantial damage to economic activity.

There are 605,102 public highway bridges (that are 20 feet in length or longer) in the U.S., as documented in the National Bridge Inventory according to the Bureau of Transportation Statistics (FHWA 2011). Much of this infrastructure is aging, a factor that contributes significantly to the degradation of bridges, as shown in Figure 7.1.

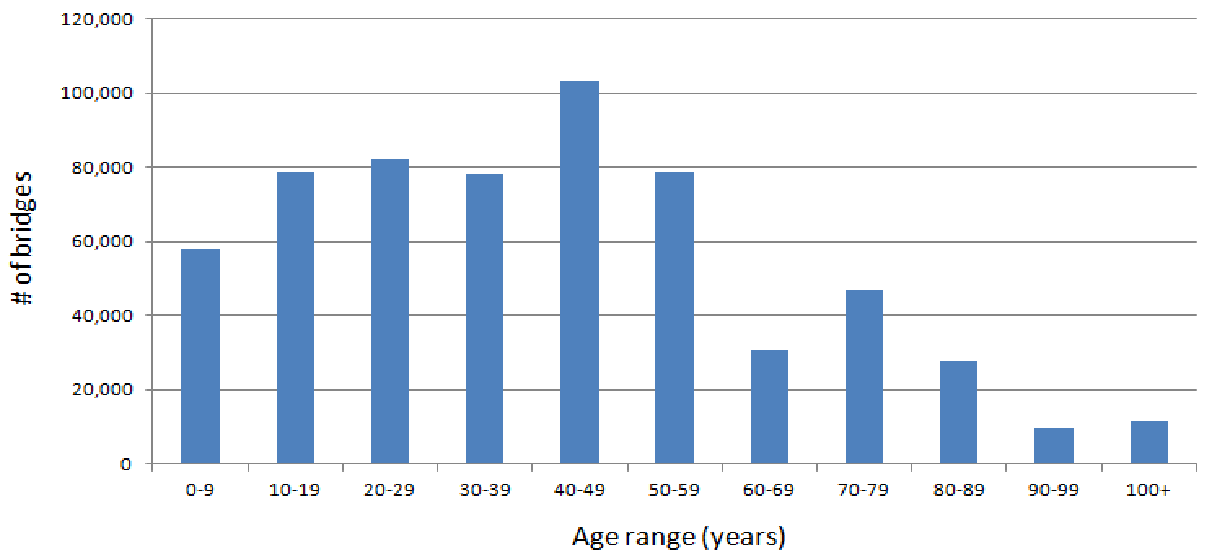


Figure 7.1: Summary of the aging U.S. DOT bridge inventory (FHWA 2011)

Bridge structures are key components of the U.S. highway system and, as mentioned previously, are vital assets for a nation's economy and security. It is therefore critical for bridge infrastructure systems to function healthily and to avoid failure. The reasons for bridge infrastructure failure are many, including the use of defective materials, deteriorating materials, substandard construction methods, improper design load ratings, natural disaster, man-made events, and long-term wear and tear. The long-term deterioration and rehabilitation of bridges is an important problem that could lead to large-scale disasters, if left unchecked.

7.1.2 Introduction to Bridge Inspections

Bridges in the U.S. are subject to the National Bridge Inspection Standards (NBIS), and are typically under the jurisdiction of the Departments of Transportation (DOTs) in the states where they are located (Moore 2011). State DOTs are generally responsible for the regular inspection and maintenance of the bridges under their jurisdiction. The Michigan Department of Transportation (MDOT), for example, employs more than 20 full-time inspectors who have specialized training and who often work in teams of two. MDOT inspectors regularly evaluate bridges and assign condition ratings using the National Bridge Inspection Rating Scale based on NBIS (MDOT 2011, Farrar 2008).

The DOTs are required to inspect, assess, and report the condition of bridge infrastructure to the FHWA. The FHWA classifies bridge infrastructure into three categories based on bridge health: 1) healthy, 2) structurally deficient, and 3) functionally obsolete. Estimates suggest that over 25% of bridges in the U.S. are currently deficient, falling into the structurally deficient or functionally obsolete categories as shown in Figure 7.2 (FHWA 2011).

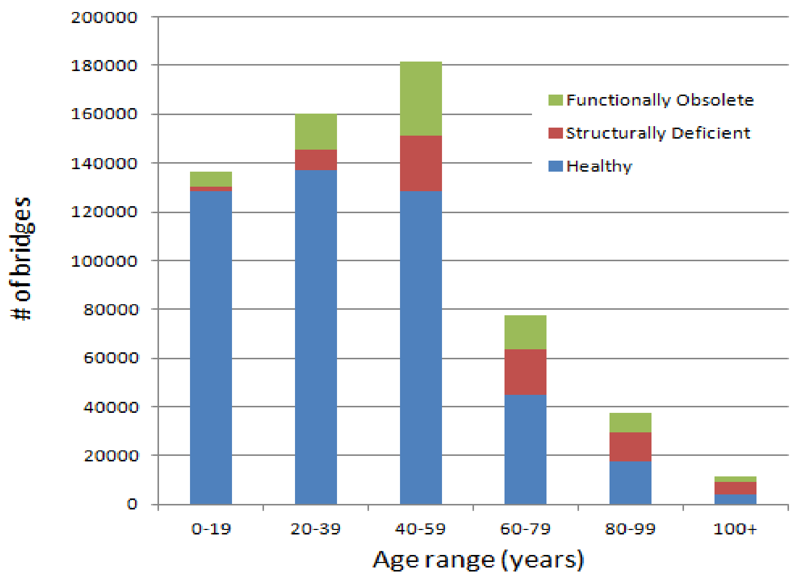


Figure 7.2: Summary of the U.S. DOT bridge infrastructure health (FHWA 2011)

Traditionally, bridge inspections have been documented manually—the bridge inspector visually assesses the condition of a bridge based on standard rating guidelines and previous bridge inspection reports (Farrar 2008). The inspector carries rating guidelines and previous reports in their paper form along with the current report form while conducting the inspection. In many cases, the inspector has to come to the site with significant preparation, and plenty of time and effort is consumed in searching, organizing, and retrieving relevant information. Upon completing the inspection and returning to the office, the relevant data gathered on the field is entered into a database management system.

In recent years, the use of bridge inspection reporting software has been explored by several state DOTs—such as Indiana's INDOT (Indiana State BIAS 2012), Wisconsin's WisDOT (Adams et al. 2005), and Washington's WSDOT (BridgeWorks.NET 2012) —as well as by independent asset management software developers (BridgeInspect^{TM} 2012). Bridge inspection reporting software solutions incorporate a centralized database and software installed on the inspector's tablet PC to facilitate the inspection process in the field. The bridge inspection software consists of interactive forms corresponding to specific components of the bridge being assessed. These forms can retrieve customized inspection guidelines and relevant historic bridge inspection data, capture bridge evaluation data, and automatically associate the captured information with the bridge components making the bridge inspection documentation intuitive.

7.1.3 Introduction to Bridge Health Monitoring

During rush hour on the evening of August 1, 2007, a steel truss bridge over the Mississippi River in Minneapolis, Minnesota, carrying the 8-lane I-35W facility experienced catastrophic failure without warning, as shown in Figure 7.3, resulting in the death of 13 motorists and injuring more than 100 people. A subsequent report by the National Transportation Safety Board identified that the probable cause of the collapse of the I-35W bridge was the inadequate load capacity, due to a design error of the gusset plates, which failed under a combination of 1) substantial increases in the weight of the bridge, which resulted from previous bridge modifications; and 2) the traffic and concentrated construction loads on the bridge on the day of the collapse (Hao 2009; NTSB 2008).

Figure: 7.3: (Left) I-35W Bridge in Minneapolis and (right) its catastrophic failure at 6:05 PM on August 01, 2007 (Image courtesy Associated Press)

The I-35W bridge failure is not a unique case of major bridge failure in recent history. Between 1989 and 2000, over 134 bridges have partially or fully collapsed in the U.S. (Wardhana and Hadipriono 2003). Although most jurisdictions in charge of bridges have inspection and maintenance measures in place to ensure that bridges meet minimum safety standards, bridge failures still occur. One reason for the shortcomings of the current measures is that bridges are assessed primarily using visual inspections. Recent studies to quantify the reliability of visual inspection methods have shown significant variability in condition ratings assigned to bridges by professional bridge inspectors (Moore et al 2001).

To increase the reliability of condition assessment, state DOTs are adopting non-destructive evaluation technologies such as accelerometers (Roberts et al 2004), chain dragging (Hearn 2005), concrete resistivity (Reis and Gallaher 2006; Yang 2008), electromagnetic fatigue sensing (Li et al 1999, Phares et al 2009), fiber optics (Inaudi and Vurpillot 1999, Casas and Cruz 2003), ground penetrating radar, pointcloud image data (Atasoy et al. 2010; Tang and Akinci 2012), scour measurement devices (Mercado and Woodroof 2008), strain gauges (Dally and Riley 2005), ultrasonic testing (Rolander et al 2001), and video imaging data (Koch and Brilakis 2011).

In the aftermath of the I-35W bridge tragedy, bridge safety has taken on a new urgency in the United States with an increased focus on Structural Health Monitoring (SHM) (Swartz et al 2007). SHM is a non-destructive, in-situ sensing and evaluation technique that uses multiple sensors embedded in a structure to monitor and analyze the structural response in order to estimate deterioration and to evaluate its consequences regarding response, capacity, and service life (Dong and Song 2010; Housner et al 1997; Karbhari 2005; Mufti 2001). In recent years, several SHM systems (Abudayyeh et al 2011; Dong and Song 2010; Doornink et al 2004; Miyamoto 2008; Roberts et al 2003) have been developed and implemented. With the increased availability of wireless data networks, sustainable SHM systems have been developed so that pervasive sensor networks allow for centralized data collection, which allows for more efficient monitoring of multiple bridges and bridge segments across large areas (Kurata et al 2011a; Kurata et al 2011b; O'Connor et al 2012; Swartz et al 2007).

7.1.4 Objectives of the Research

Despite the proliferation of non-destructive evaluation techniques, the vast majority of bridge condition assessments are currently done using visual inspection methods. The primary aim of this research is to integrate a wireless SHM system (Kurata et al 2011a; Kurata et al 2011b; O'Connor et al 2012; Swartz et al 2007) with the visual inspection process to provide meaningful input to the inspector in order to add more objectivity to bridge condition assessment methods in current inspection processes. Bridge component ratings, with added objectivity, can then be used to allocate bridge maintenance and rehabilitation resources more efficiently.

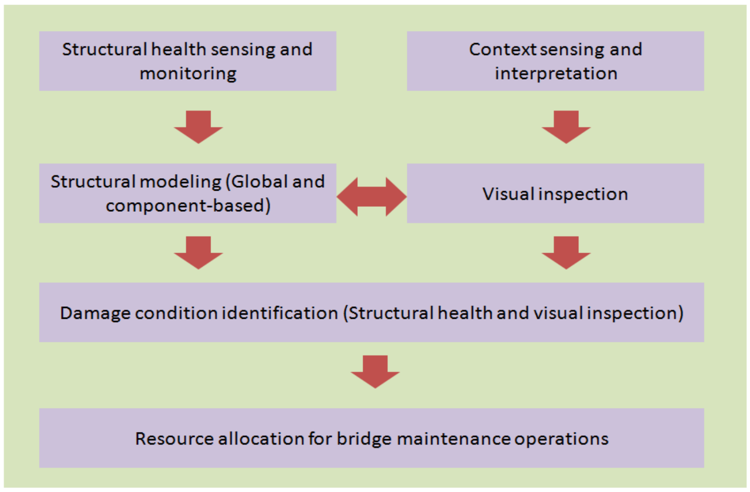
In order to achieve the aforementioned objectives, a context-aware bridge inspection solution was designed and implemented. This chapter first presents the over-arching architecture of the developed solution and the integration of the context-aware computing framework discussed in Chapter 6. The chapter then presents the cyber-infrastructure developed to support the solution. A relational bridge data model is designed to store physical, functional, and bridge condition assessment/inspection information in conjunction with the cyber-infrastructure required for supporting the SHM system adopted into the solution. The scalable and extensible bridge inspection software developed as a part of this research to query, access, retrieve, and document digital and multimedia condition assessment information is presented, and its functionality and features are then discussed. The software is extended to incorporate multiple damage condition assessments methods such as traditional visual assessment, and SHM system analytics-based assessment methods. As a proof of concept, specific examples of SHM information analytics that provide insight into the damage condition assessment of bridges are then presented. Then, the limitations, challenges, and future work that must be pursued to exploit the power of this solution are discussed. Finally, the achievements of the research effort and the advantages of the solutions developed are presented.

7.2 Technical Approach for Context-Aware Bridge Inspection

7.2.1 Overview of the Technical Approach

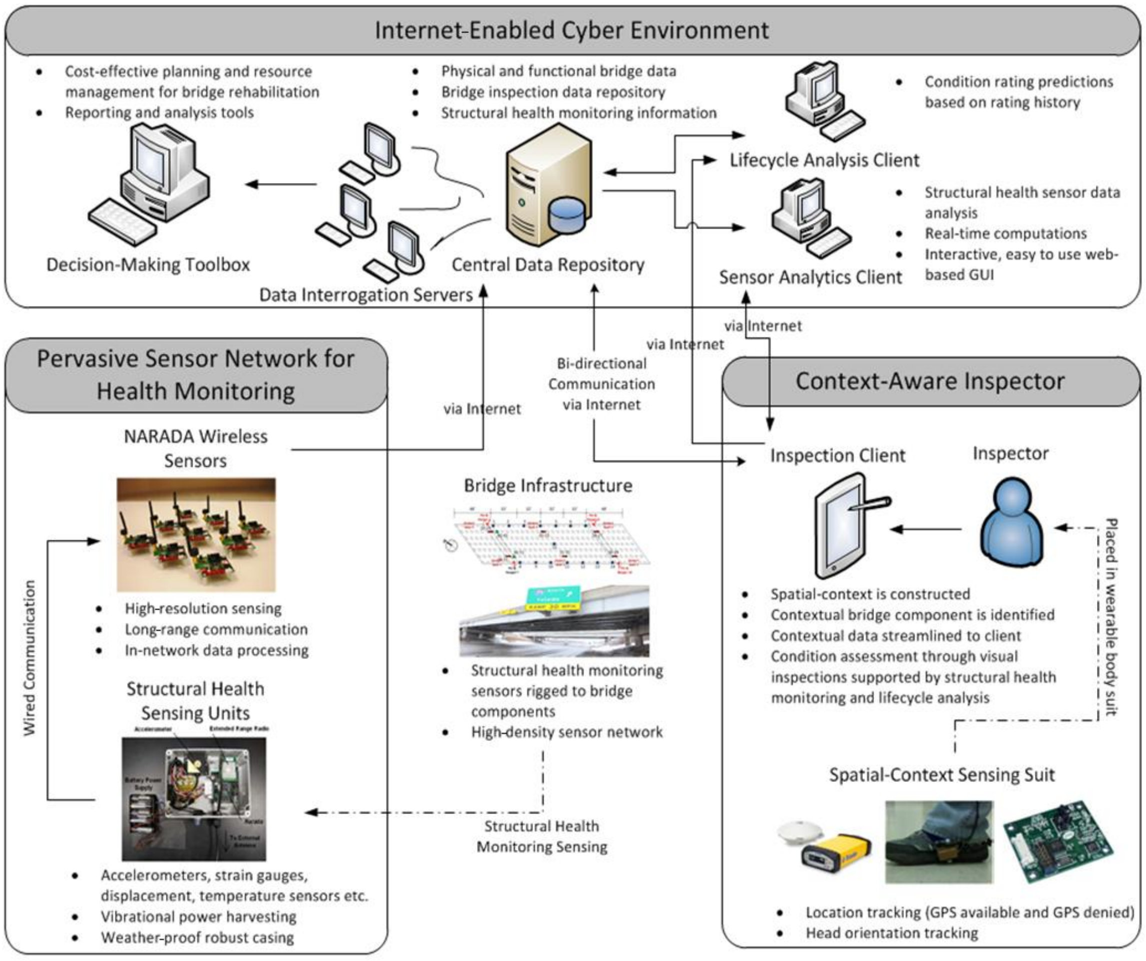
Bridge infrastructure asset management emphasizes timely actions—such as infrastructure preservation, rehabilitation, and replacement—through cost-effective planning and resource allocation. Bridge owners are often faced with the task of prioritizing operations to allocate limited resources to in competing bridge infrastructure maintenance operations. The SHM-integrated context-aware bridge inspection approach presented in this chapter was developed as part of a project to provide bridge owners with actionable information for improved decision-making. The workflow of the SHM-integrated context-aware bridge inspection process is shown in Figure 7.4.

The overview of the architecture of the SHM-integrated context-aware bridge inspection solution proposed in this chapter is shown in Figure 7.5. The developed solution is scalable and extensible to most highway bridges and is deployed using the northbound I-275 highway bridge crossing over Telegraph Road (US-24) in Monroe, Michigan as the case study bridge (CSB) and as a proof of concept of the inspection system developed in this research.



Structural health sensing and monitoring
Context sensing and interpretation
Structural modeling (Global and component-based)
Visual inspection
Damage condition identification (Structural health and visual inspection)
Resource allocation for bridge maintenance operations

Figure 7.4: Workflow of resource allocation for bridge maintenance operations



Internet-Enabled Cyber Environment
Cost-effective planning and resource management for bridge rehabilitation
Reporting and analysis tools
Physical and functional bridge data
Bridge inspection data repository
Structural health monitoring information
Condition rating predictions based on rating history
Lifecycle Analysis Client
Structural health sensor data analysis
Real-time computations
Interactive, easy to use web-based GUI
Decision-Making Toolbox
Central Data Repository
Sensor Analytics Client
Data Interrogation Servers
via Internet
via Internet
Bi-directional Communication via Internet
via Internet
Pervasive Sensor Network for Health Monitoring
NARADA Wireless Sensors
High-resolution sensing
Long-range communication
In-network data processing
Wired Communication
Structural Health Sensing Units
Accelerometers, strain gauges, displacement, temperature sensors etc.
Vibrational power harvesting
Weather-proof robust casing
Bridge Infrastructure
Structural health monitoring sensors rigged to bridge components
High-density sensor network
Structural Health Monitoring Sensing
Context-Aware Inspector
Inspection Client
Inspector
Spatial-context is constructed
Contextual bridge component is identified
Contextual data streamlined to client
Condition assessment through visual inspections supported by structural health monitoring and lifecycle analysis
Placed in wearable body suit
Spatial-Context Sensing Suit
Location tracking (GPS available and GPS denied)
Head orientation tracking

Figure 7.5: The overview of SHM integrated context-aware bridge inspections

The crossing is a skewed, steel-concrete composite bridge with three spans, and is shown in Figure 7.6. The bridge is supported by two abutments at the ends and two sets of concrete piers at the middle. The total length of the bridge is 223 ft, and the middle span—which is also the longest span—is 140 ft in length. The bridge superstructure is comprised of 7 steel girders that support an 8-inch concrete deck slab. The girders are simply supported at the piers and abutments, and each girder is suspended by two pairs of pin and hanger assemblies in the middle span. The middle span deck is connected to the end spans by an expansion joint and a construction joint aligned directly above the pin and hanger assemblies.

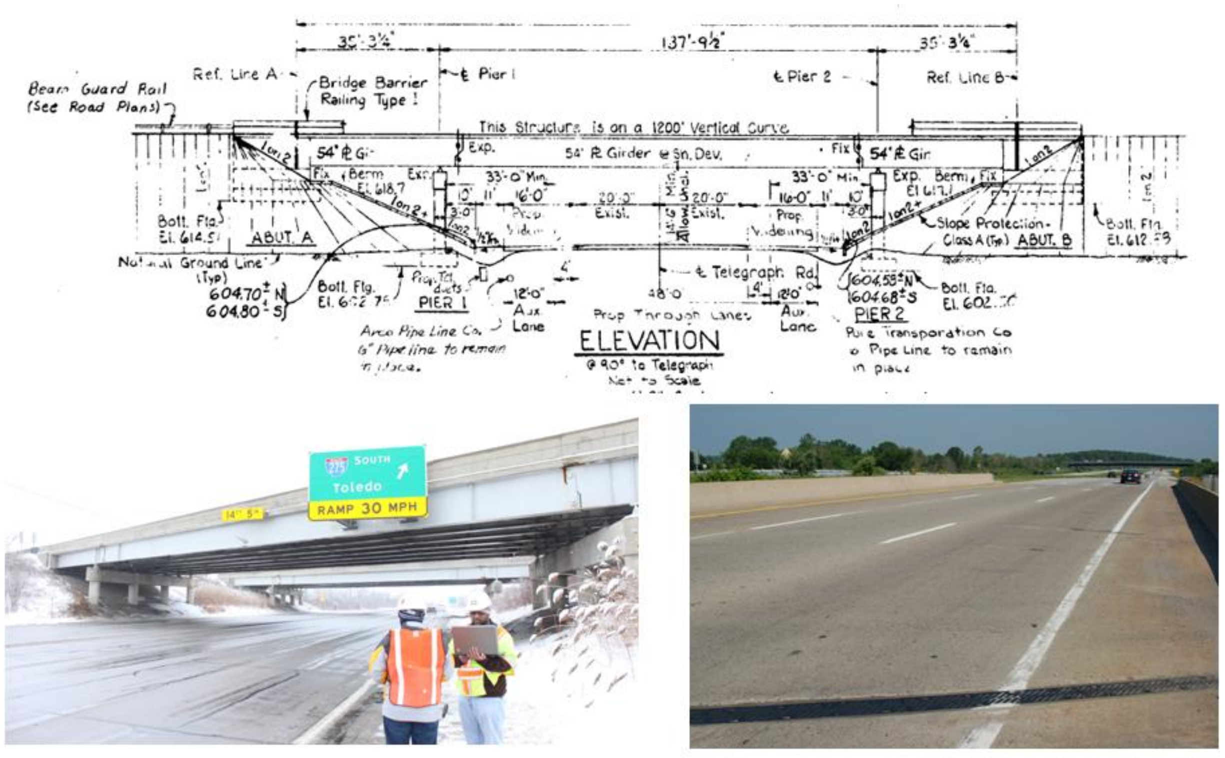


Figure 7.6: Northbound I-275 crossing over Telegraph Road (US-24)

The CSB was recommended by MDOT for two primary reasons: 1) it is the best topological and ontological representative of most highway bridges; and 2) it is comprised of significantly diverse components found in a typical Michigan highway bridge.

7.2.2 Central Data Repository

The Central Data Repository (CDR) is implemented as a PostgreSQL database with a multi-platform client-server application for storing data. PostgreSQL is an open source object-relational database management system. The multiplatform client-server application is developed using the Internet Communications Engine (Ice) middleware, and can be accessed through a number of development environments, including Python, Java, C++, and C#. The CDR stores physical and functional characteristics of bridge infrastructure along with structural health monitoring and bridge inspection information, as shown in Figure 7.7.

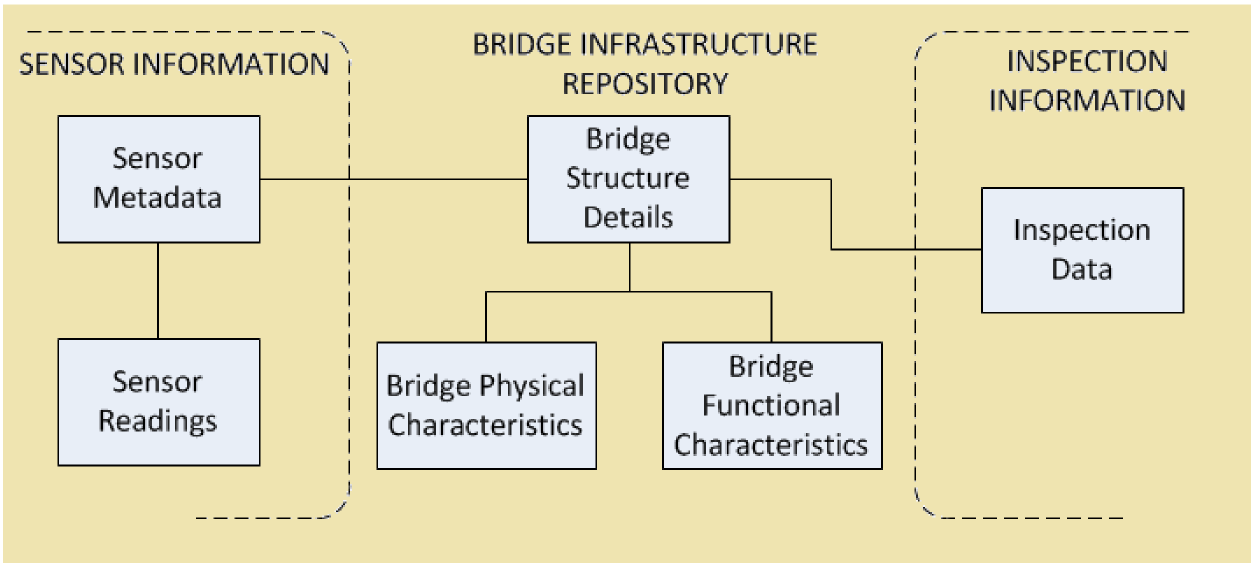


Figure 7.7: Overview of the data stored in the CDR

The CDR stores three main categories of information for bridge infrastructure fleets: 1) bridge information, 2) sensor information, and 3) inspection information. Bridge design information includes information regarding bridge design (such as identity, location, desired inspection frequency, and structural components), physical characteristics (such as component material and geometry), and functional characteristics (such as finite element models, traffic volume, and load distributions) of the bridges in the database. Sensor information includes sensor metadata (information about where/how sensors are deployed, the types/characteristics of sensors deployed, etc.) and the sensor readings themselves (van der Linden et al 2013). Inspection information includes bridge inspection data along with accompanying digital and multimedia data that supports assessment decisions. Condition assessment based on visual inspections and SHM are conducted using bridge information along with inspection information and sensor information, respectively. The relational model of the database from visual inspection condition assessment perspective is presented in detail in Section 7.3.1. The relational model of the database from the SHM condition assessment perspective (van der Linden et al 2013) is beyond the scope of this chapter.

7.2.3 Pervasive Sensor Network for Structural Health Monitoring

Automated wireless pervasive sensor networks are used for SHM of bridge infrastructure. The wireless SHM system features access to the internet to facilitate communication with the CDR and the rest of the cyber-infrastructure environment. The pervasive sensor network is designed to be reliable with enhanced communication ranges and long-term power harvesting strategies, and to operate on solar energy (Kurata et al 2011a; Kurata et al 2011b; Swartz et al 2007). Communication stability and system robustness were verified at the CSB through the short-term deployment of dense wireless sensor networks at various locations on the bridge. Permanent wireless sensors were then deployed for the long-term continuous monitoring of SHM data.

The architecture of the wireless SHM system is designed with system functionality and power usage delineated to different hierarchical tiers. Sensors that measure vibration responses, atmospheric profiles, strain hysteresis, etc. are tethered to low-power *Narada* wireless sensor nodes that collect data and process it in-network to compress the information to be communicated and stored by the system. The processed data is transmitted by the *Narada* sensors to the CDR within a server. The CDR is accessible through the internet-enabled cyber-environment to a series of data interrogation servers hosting modeling tools, damage-detection tools, and system identification tools (Kurata et al 2011a; Kurata et al 2011b; O'Connor et al 2012).

As a proof of concept, the underside of the CSB deck was selected for the monitoring and long-term deployment of *Narada* wireless sensor nodes. *Narada* wireless sensors are magnetically mounted to the surface of the steel girders, hangers, and the deck. These *Narada* nodes are powered through solar energy harvesting systems, each consisting of a 3.3W solar panel, a low-power energy charging circuit board, and a rechargeable battery pack. The data logging system installed at the *Narada* receiver station is an industrial-grade single board computer running Linux, and it is designed for embedded, low-power applications permitting fan-less operation over a temperature range from -40^{o}C to 85^{o}C. The system accesses the internet through a 3G mobile phone network, grants users with remote access, and is designed for robust operations with automated rebooting. The *Narada* server program starts automatically after the system starts up, initiates data collection from the *Narada* sensors, collects the appropriate data, and triggers the sleep mode until the next scheduled data collection period. This long-term SHM system has been running continuously since September 2011, and had additional *Narada* sensor nodes incorporated into the system from May through December of 2012.

7.2.4 Context-Aware Inspector

During an inspection routine, the inspector visually assesses the condition of the components that comprise the bridge. Based on a priori knowledge of the bridge components' design hierarchy, the inspector navigates the software to the form corresponding to the component, and reports the assessed condition. Context-aware computing offers the possibility of making inspections more efficient by reducing the time required to navigate inspection software and the effort spent by inspectors to learn, remember, and recall the design hierarchy and ontology of the components comprising bridges. The inspector carries a tablet PC, running a Windows operating system, with access to the internet through a 3G mobile network. The tablet PC allows the inspector to navigate using a stylus, recognizes intelligible handwriting input from the touch screen, and has inbuilt features to record audio-visual multimedia. A context-aware computing application and an innovative bridge inspection software known as Contextual Object Identifier (CONOID) and Bridge Inspection Toolkit (BIT), respectively, are installed on the tablet PC.

CONOID identifies the component of interest to the inspector and is developed using the context-aware computing framework presented in Chapter 6. The CONOID application actively senses spatial contextual parameters and reconciles sensed context with bridge component geometry to predict the component of interest. CONOID communicates the predicted component of interest to the BIT, thus allowing the BIT to automatically query the CDR to retrieve information relevant to the component being assessed.

The spatial contextual parameters sensed by CONOID are the inspector's location and head orientation to ascertain the inspector's line of sight, as described in Section 6.2.2. The location is sensed using the Global Positioning System (GPS) where available, and the Ultra-Wide Ban (UWB) positioning system in the underside of the superstructure where GPS is unavailable. The inspector's head orientation is sensed using either a magnetic compass or non-magnetic orientation trackers with accelerometers/gyroscopes. CONOID uses the sensed spatial contextual parameters to construct the spatial-context of the inspector, as described in Section 6.2.3. The CDR contains information about the bridge components' geometry, as described in Section 7.3.1. The CONOID application temporarily downloads the component geometry information relevant to the bridge being assessed. The constructed spatial context and the component geometry information are used as inputs to CONOID's containment and prioritization algorithms, which are described in Sections 6.2.7 and 6.2.8, respectively. The containment algorithm identifies the components in the inspector's field of view, and the prioritization algorithm predicts the component of interest to the inspector.

The predicted component of interest is then passed to the BIT to query for contextually relevant information including component condition history, supporting documentation, rating guidelines, and SHM analysis. The component condition history, supporting documentation, and rating guidelines are retrieved from the CDR whereas the SHM analysis is retrieved by querying the sensor analytics client described in Section 7.2.5. The inspector visually inspects the component and reconciles field observations with the retrieved contextually relevant information to assess the condition of the component. The assessed condition is documented in the CDR along with any supporting digital and multimedia documentation. The features and functionality supported by the BIT are presented in Section 7.3.2.

7.2.5 Sensor Analytics Client

The BIT allows bridge inspectors to access meaningful information regarding the structural health of bridge components that are of interest to them. As mentioned previously, the structural health bridge components is monitored using a pervasive sensor network that stores sensor readings and metadata in the CDR. The sensor analytics client processes raw sensor data to generate a meaningful interpretation of structural health that is easily understood by the inspector.

The sensor analytics client is hosted on a machine, running Linux, with access to the sensor data stored in the CDR through the internet. Bridge components that are monitored using the pervasive sensor network may have one or more types of SHM analysis that can be performed to assess their condition. The client is comprised of a set of Python scripts, each corresponding to a type of SHM analysis. The appropriate Python scripts are invoked by querying the sensor analytics client with the component of interest and the desired type SHM analysis. Upon being invoked, the Python script connects to the CDR and retrieves the raw sensor data to process the underlying SHM algorithms to generate the requested analysis. The generated analysis is written to a Google Chart object, using the Google Chart API, and is embedded in a webpage that is returned to the BIT.

As a proof of concept, SHM analysis algorithms were developed for deck cracking, ping and hanger fatigue, pin and hanger lock-up, and deck-girder system composite action analysis. The theoretical concepts of the aforementioned SHM analysis and their interpretations in terms of bridge condition assessment are described in detail in Section 7.4. The developed SHM analysis algorithms were tested and validated using the sensor data collected from the pervasive sensor network installed on the CSB.

7.2.6 Decision Making Toolbox

Infrastructure ownership, being a highly fragmented sector, lacks any collaboration and standardization of decision-making data collection and analysis tools (Ettouney and Alampalli 2007). Bridge owners make infrastructure maintenance decisions based on a mixture of facts, experience, accumulated knowledge, and rules of thumb (Alampalli and Ettouney 2010). In recent years, decision-making methods and tools targeted toward bridge owners have moved toward automation and computerization (Flintsch and Bryant Jr. 2006). As a result, bridge owners have turned to software solutions for bridge inspection reporting and decision-making analysis.

Bridge owners use computer programs, available from various sources, for analysis and decision-making. These systems typically rely on cumbersome and unintuitive processes to conduct analysis and develop reports, thus rendering them difficult to use (Alampalli and Ettouney 2010). Pontis is one such bridge management software developed by Cambridge Systematics and Optima Inc. for the FHWA (FHWA 2008). Pontis stores bridge infrastructure inventory and inspection information in a relational database to develop maintenance budgets and programs. However, unlike the BIT, Pontis does not incorporate SHM data storage and analysis to assess bridge component condition in order to make appropriate maintenance decisions.

An innovative software solution known as Decision Making Toolbox (DMT) uses the data collected through bridge inspection and SHM to support cost-effective planning and resource allocation decisions for timely bridge maintenance action including preservation, rehabilitation, and replacement (Alampalli and Ettouney 2010). DMT is a coherent, comprehensive set of analysis programs implemented to run on a Windows operating system. DMT also provides a medium to develop rapid application analysis through a visual modeling environment without any programming (Alampalli and Ettouney 2010). DMT uses multiple algorithms including modal identification, parameter identification, and rain flow techniques in conjunction with inspection and SHM data to compute, in real-time, different reliabilities and risk estimates for bridge components. Moreover, capacity, demand, and failure consequences of bridge components are also estimated and summarized in an elegant form for bridge decision-makers to use (Alampalli and Ettouney 2010).

7.3 Bridge Inspection Software

7.3.1 Bridge Inspection Data in the Central Data Repository

The BIT communicates bi-directionally with the infrastructure inventory and inspection information present in the CDR database while accessing SHM information through the sensor analytics client. The entity-relationship model shown in Figure 7.8 describes the part of the database that directly interacts with the BIT and that does not describe the rest of the database.

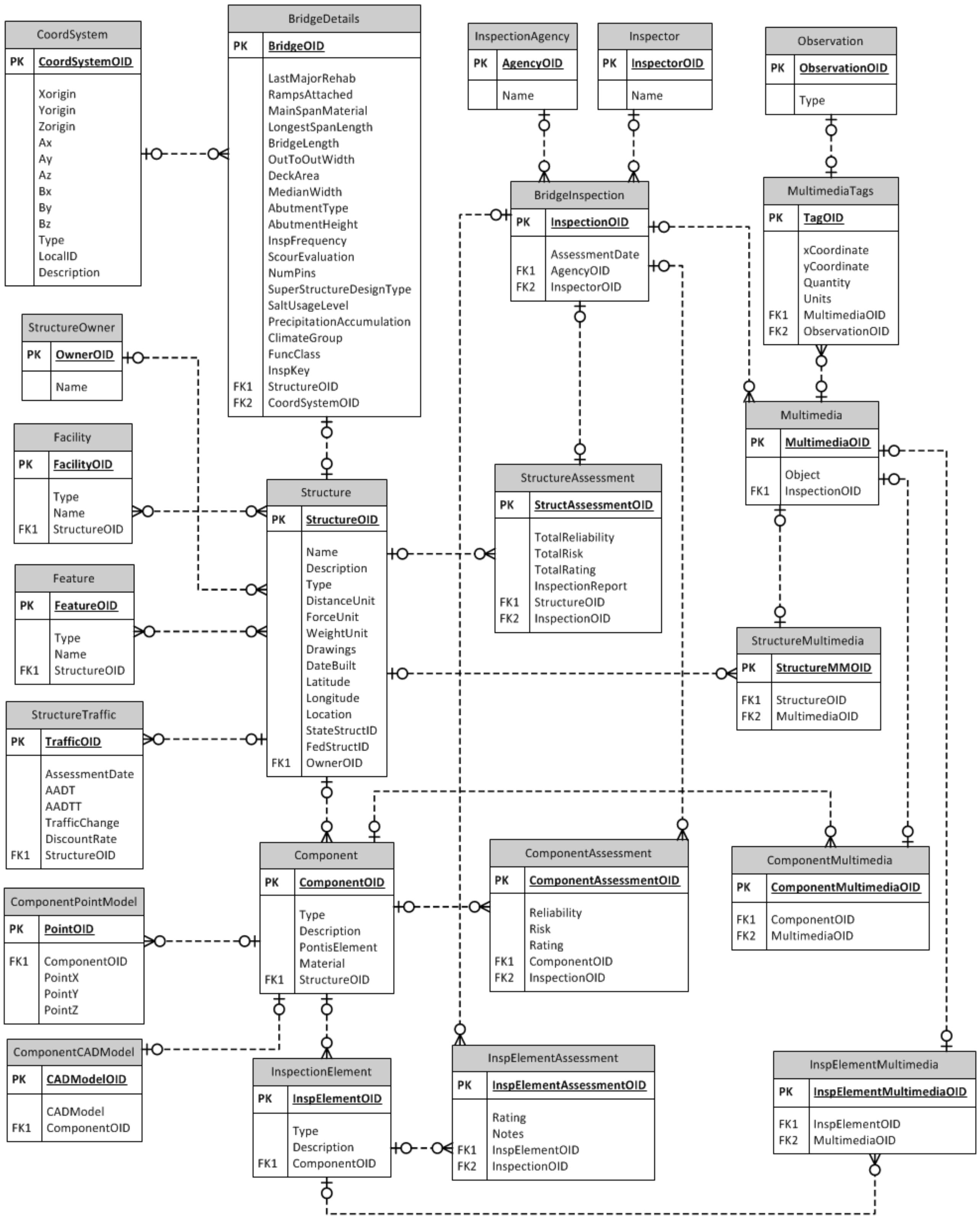


Figure 7.8: Entity-relationship model of the CDR database that directly interacts with the BIT

The database stores inventory and geometry information about bridge structures and their constituent physical components as illustrated in Figure 7.9. The CSB, for example, is comprised of 3 deck components (1 deck per span), 1 strip-seal expansion joint, 1 pourable joint, 21 girders (7 girders per span), 14 pin and hanger assemblies (connecting 7 girder pairs at 2 span joints), 8 concrete piers, 2 concrete column caps, 2 abutments, 14 movable rocker bearings (supporting the 7 main span girders at the 2 column caps), 2 approach slabs, and 2 deck railing components. The bridge's native Cartesian coordinate system is defined by describing its origin and axes in the global coordinate system (Latitude, Longitude, and Altitude). Component geometry is stored as CAD models and/or point clouds that are registered to the native coordinate system of the bridge.

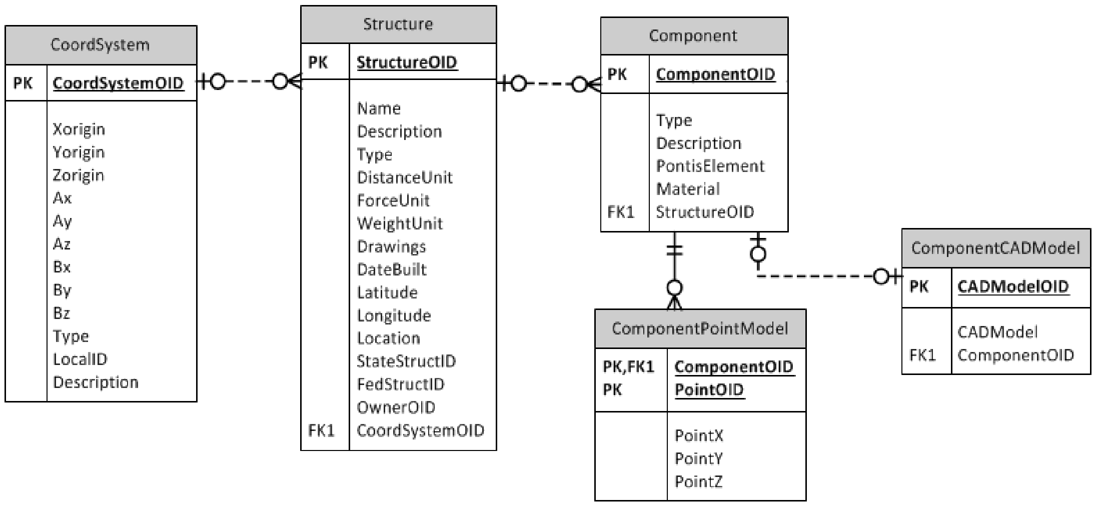


Figure 7.9: Entity-relationship model of the bridge infrastructure inventory and constituent component geometry

Each component's condition is assessed by observing certain elements of the component, and these elements are stored as illustrated in Figure 7.10. For example, the CSB's deck components are assessed by observing the deck surface, sidewalk, structure inventory and appraisal (SIA), and drainage elements. Girders are assessed by observing stringer SIA and section loss elements.

Abutments are assessed on the SIA and slope protection aspects, whereas approach slabs are assessed on the pavement and shoulder elements. Pin and hanger assemblies, piers, pier caps, expansion joints, pourable joints, rocker bearings, and railings are assessed in their entirety.

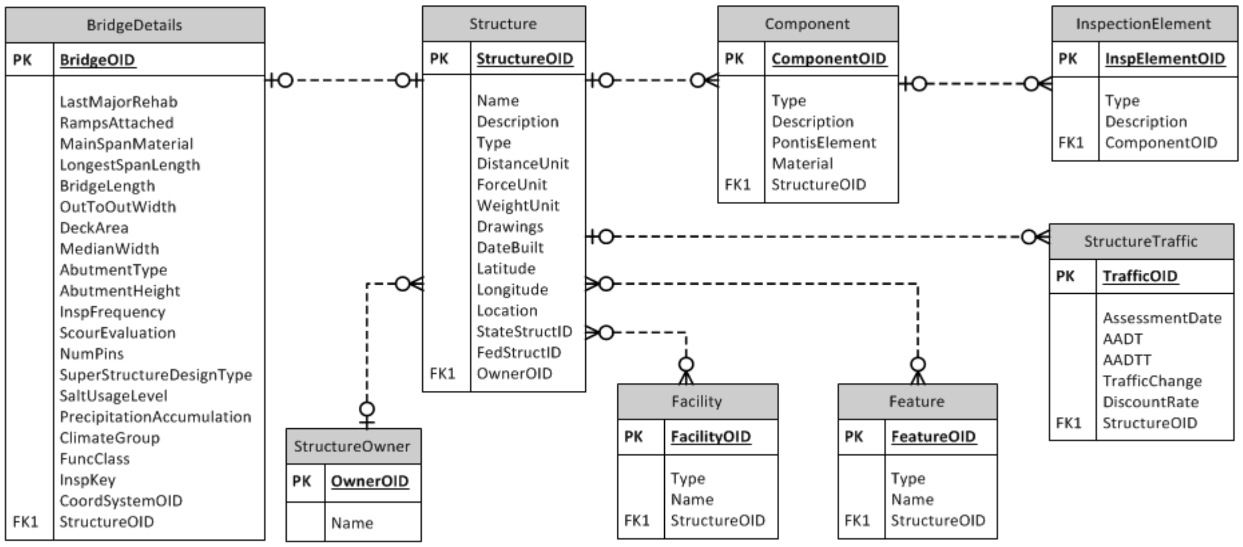


Figure 7.10: Entity-relationship model of the functional characteristics of the bridges

The database documents information regarding owner organizations and the bridge inventory under their jurisdiction. Also documented are the operating facilities (highways, railway lines, and roads) carried by the bridges and the features (natural and man-made features such as rivers, valleys, highways, and railway lines) underneath the bridges. As illustrated in Figure 7.10, the characteristics of traffic loads experienced by the bridge are documented using transportation engineering measurements including annual average daily traffic (AADT) and annual average daily truck traffic (AADTT). The non-traffic-related characteristics of the bridge—including recommended inspection frequency, superstructure design type, salt usage level, precipitation, climate group—are also archived. Bridge characteristics combined with traffic volume information are used to establish similarity among bridges while comparing equivalent bridges during SHM and inspection analysis. The structural characteristics of the bridge inventory are stored in the database as finite element models (FEM). The FEM data is also registered to the bridge's native coordinate system to maintain concurrence with geometric models. However, FEM data structures are not discussed in detail as the BIT does not interact directly with FEM.

The condition assessment information and multimedia documentation that supports assessment decisions are stored, as illustrated in Figures 7.11 and 7.12, respectively. The database keeps a log about trained inspection personnel and inspection agencies operating across infrastructure jurisdictions. The condition ratings corresponding to the assessed elements, components, and bridge inventory are documented along with the inspection metadata.

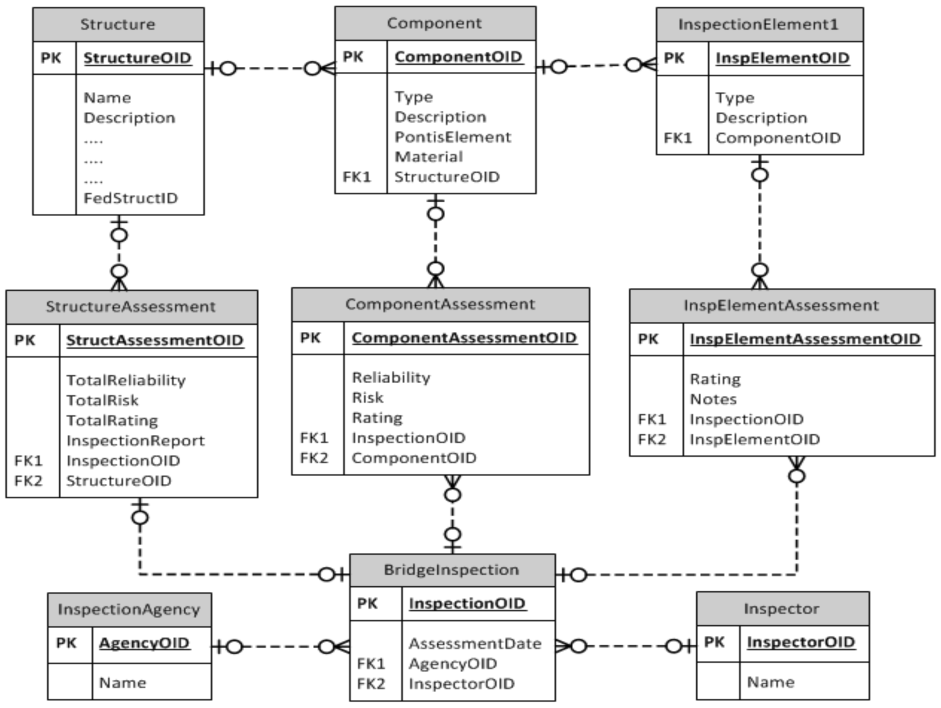


Figure 7.11: Entity-relationship model of inspections and condition assessment information

The database also allows users of the BIT to archive multimedia—such as text notes, images, audio clips, and videos—taken during an inspection routine by associating captured multimedia with appropriate elements, components, and bridge structures. The database also allows images from multimedia to be tagged with the type and quantity of damage observed during inspections for reference.

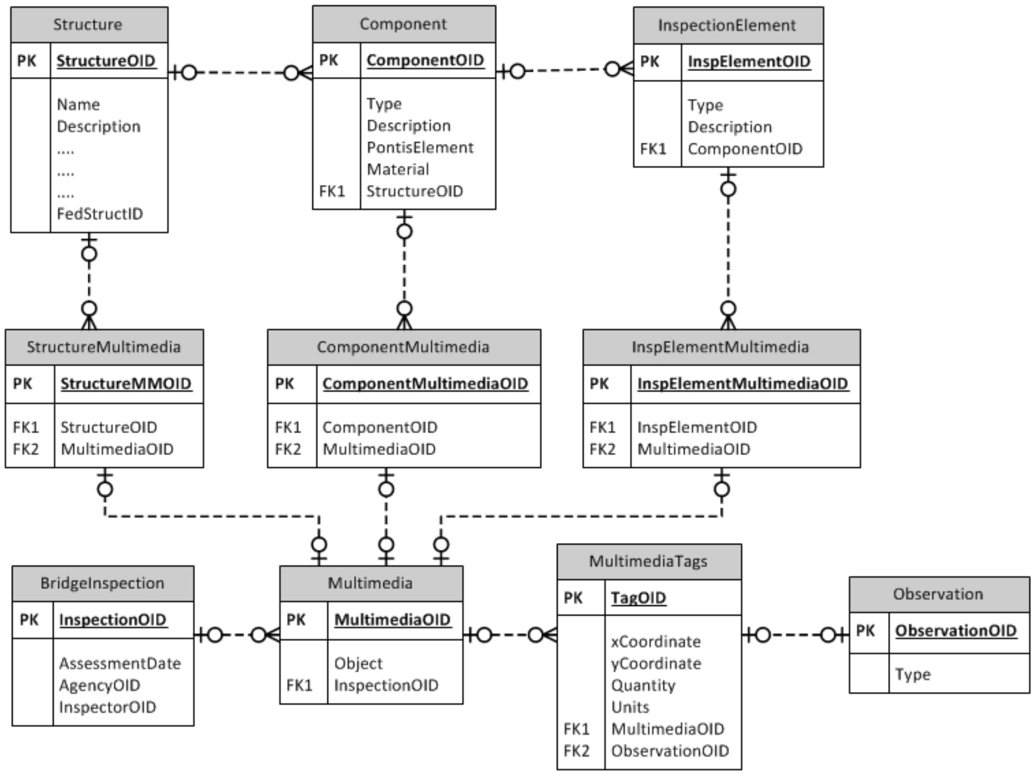


Figure 7.12: Entity-relationship model of multimedia documentation to support condition assessment decisions

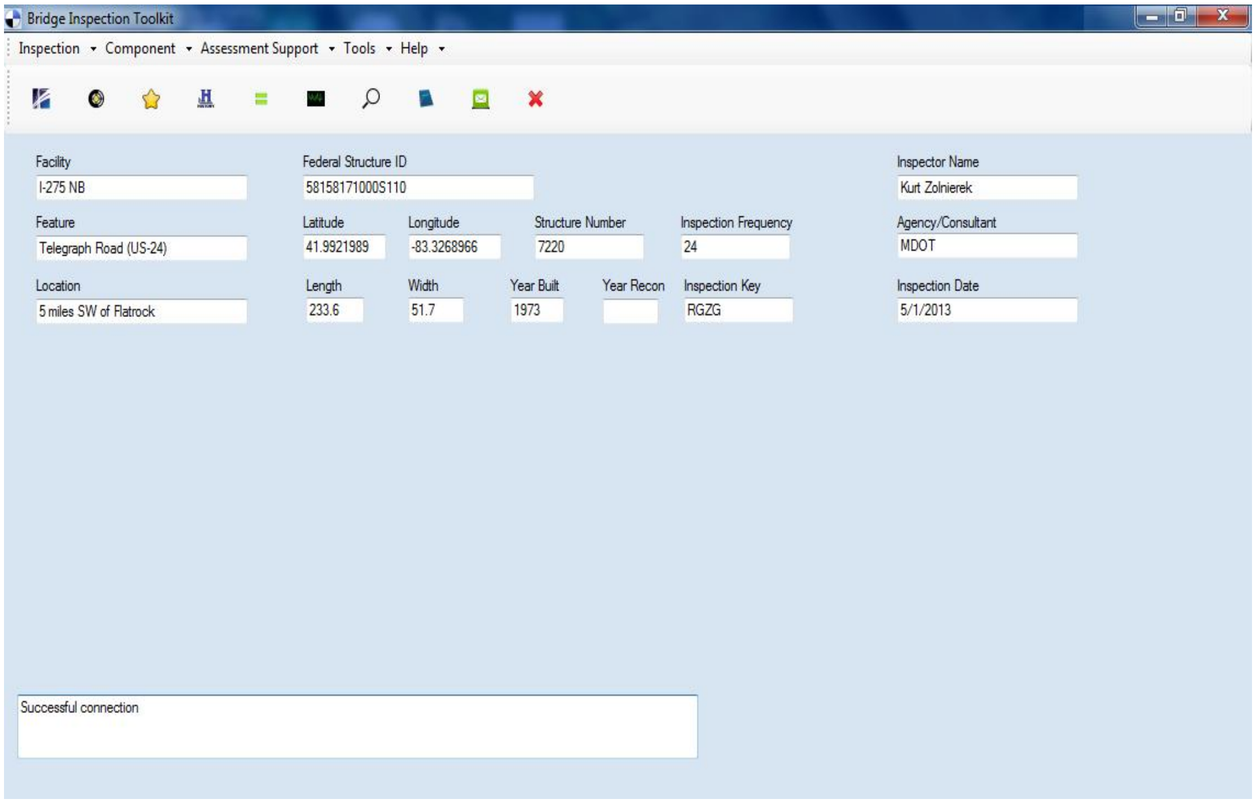
7.3.2 Bridge Inspection Toolkit

As mentioned previously, traditional inspection methods involve collecting condition assessment data in paper form during the inspection, and entering the relevant data gathered into a database management system after returning to the office. This process involves spending significant time and effort in organizing large amounts of information in the pre-inspection, on-site, and post-inspection phases of the bridge condition assessment operation.

The BIT is an innovative proprietary software solution that takes a new and unique approach for intelligent condition assessment decision-making. The BIT is written in the C# programming language using the WinForms API included as a part of Microsoft’s .NET Framework and is implemented to run under the Windows operating system. The BIT is a scalable, extensible, and modular software tool that not only allows inspectors to access and document condition assessment information, but that also provides decision-making support by seamlessly integrating context-awareness and SHM analysis tools. The BIT tutorial is presented in Appendix D.

7.3.2.1 BIT Interface and Functionality

Inspectors are first asked to log in using assigned credentials to access the BIT software. Once authenticated, the inspector is allowed to select the bridge under inspection by choosing from the list of bridges under the jurisdiction of the inspector’s agency. For each bridge, the inspector can use the BIT in either the inspection mode or the exploratory mode. In the exploratory mode, the inspector is allowed to access previously documented inspection data but is prohibited from adding new inspection data and editing previously existing data. When an inspector starts a new inspection for a bridge, the BIT takes the inspector to the main inspection window, which is shown in Figure 7.13. The top portion of the window displays the inspection metadata and the functionality available to the inspector during the operation. The bottom portion of the main window has empty space that can be used to dock and/or anchor forms that facilitate the execution of the BIT functionality. The BIT functionality includes tools to select the navigation method, access component assessment history, assess component condition, document condition assessment, track inspection progress, and print appropriately formatted inspection reports.



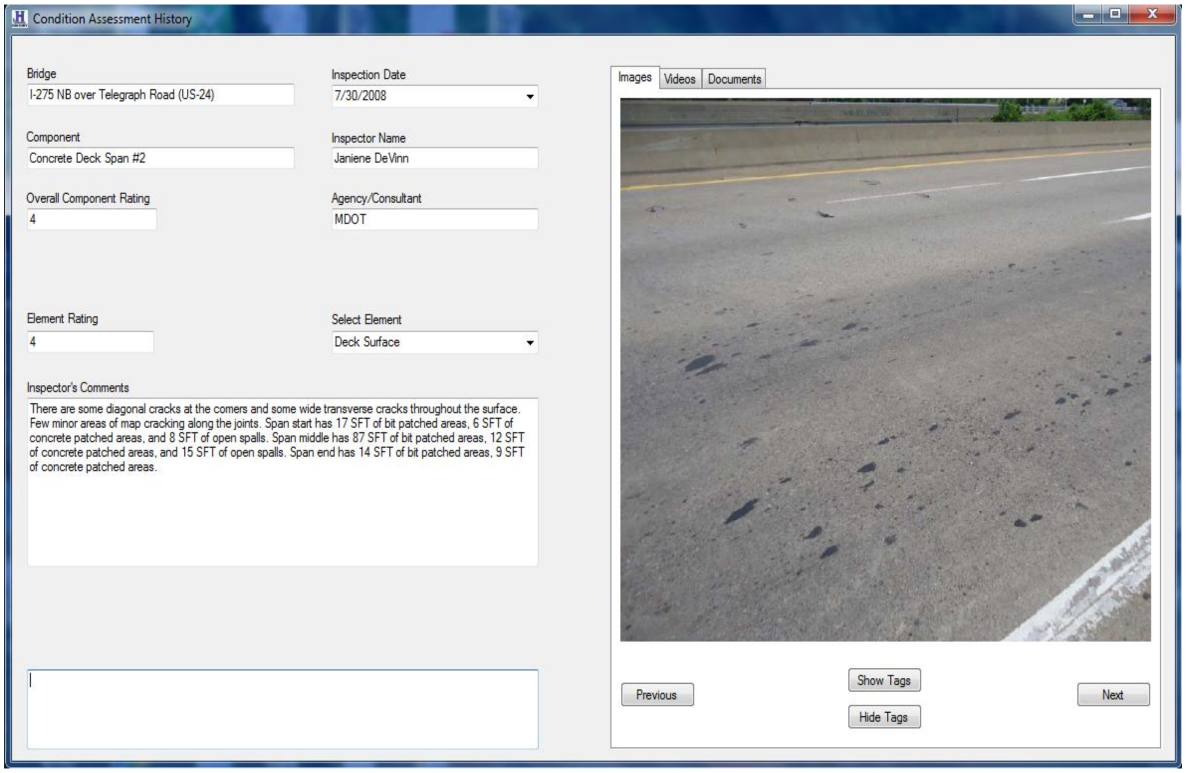
Facility: I-275 NB	Federal Structure ID: 58158171000S110			Inspector Name: Kurt Zolnierek
Feature: Telegraph Road (US-24)	Latitude: 41.9921989	Longitude: -83.3268966	Structure Number: 7220	Inspection Frequency: 24
Agency/Consultant: MDOT				
Location: 5 miles SW of Flatrock	Length: 233.6	Width: 51.7	Year Built: 1973	Year Recon:
Inspection Key: RGZG	Inspection Date: 5/1/2013			

Successful connection

Figure 7.13: The graphical user interface (GUI) of the BIT showing the main inspection window

The inspector is allowed to select the mode of navigating the BIT software by choosing either the manual mode (default) or the context-aware mode. In the manual mode, the task of selecting the component of interest and navigating the software is completely incumbent on the inspector. By selecting the context-aware mode, the inspector delineates the task of navigating the software to the CONOID application that predicts the inspector’s component of interest (the component being inspected), confirms it with the inspector, and hands over the component of interest to the BIT.

Upon query, the inspector can access the component's assessment history functionality presented in an easy to use, interactive form as shown in Figure 7.14. The form allows the inspector to intuitively view the history of condition ratings, inspector notes, supporting multimedia documentation, and assessment metadata across all elements associated with the component. The inspector can navigate across elements and assessment history by selecting desired parameters from the dropdown boxes provided. The inspector can view any available multimedia supporting historic condition assessment by clicking across tabs (differentiating multimedia by type) and using the provided navigation buttons.



Bridge: I-275 NB over Telegraph Road (US-24)

Inspection Date: 7/30/2008

Component: Concrete Deck Span #2

Inspector Name: Janiene DeVinn

Overall Component Rating: 4

Agency/Consultant: MDOT

Element Rating: 4

Select Element: Deck Surface

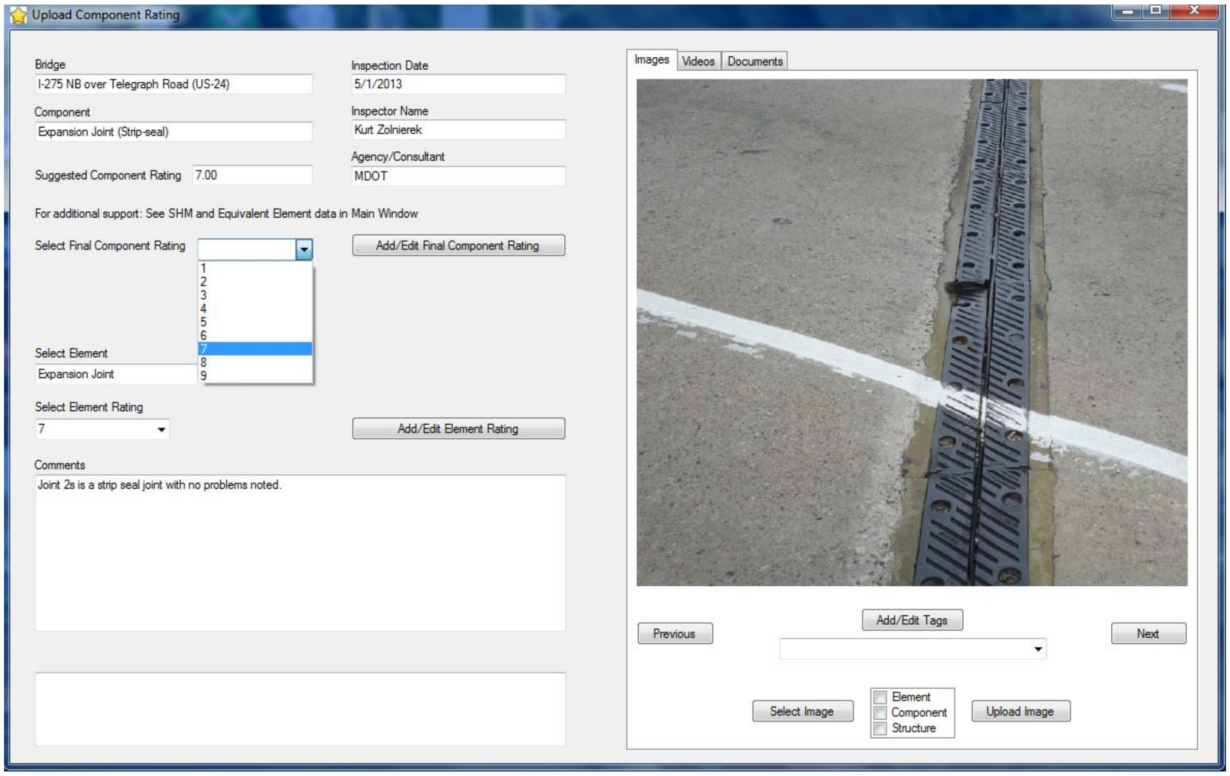
Inspector's Comments: There are some diagonal cracks at the comers and some wide transverse cracks throughout the surface. Few minor areas of map cracking along the joints. Span start has 17 SFT of bit patched areas, 6 SFT of concrete patched areas, and 8 SFT of open spalls. Span middle has 87 SFT of bit patched areas, 12 SFT of concrete patched areas, and 15 SFT of open spalls. Span end has 14 SFT of bit patched areas, 9 SFT of concrete patched areas.

Figure 7.14: BIT GUI showing a component's condition assessment history

The inspector uploads condition assessment information using the functionality provided, as shown in Figure 7.15. The functionality allows the inspector to rate the condition of the elements corresponding to the contextual-component on the NBIS scale (MDOT 2011, Farrar 2008) by selecting the appropriate rating from the dropdown box. The functionality also allows the inspector to add and edit notes using the provided text box. Multimedia supporting condition assessment decisions are added by selecting desired files from the camera or disk and associating them with the element and/or component and/or bridge being inspected. Image multimedia is geo-tagged with standardized condition observations by selecting the type of observation and by clicking on the portion of the image depicting the observed condition.

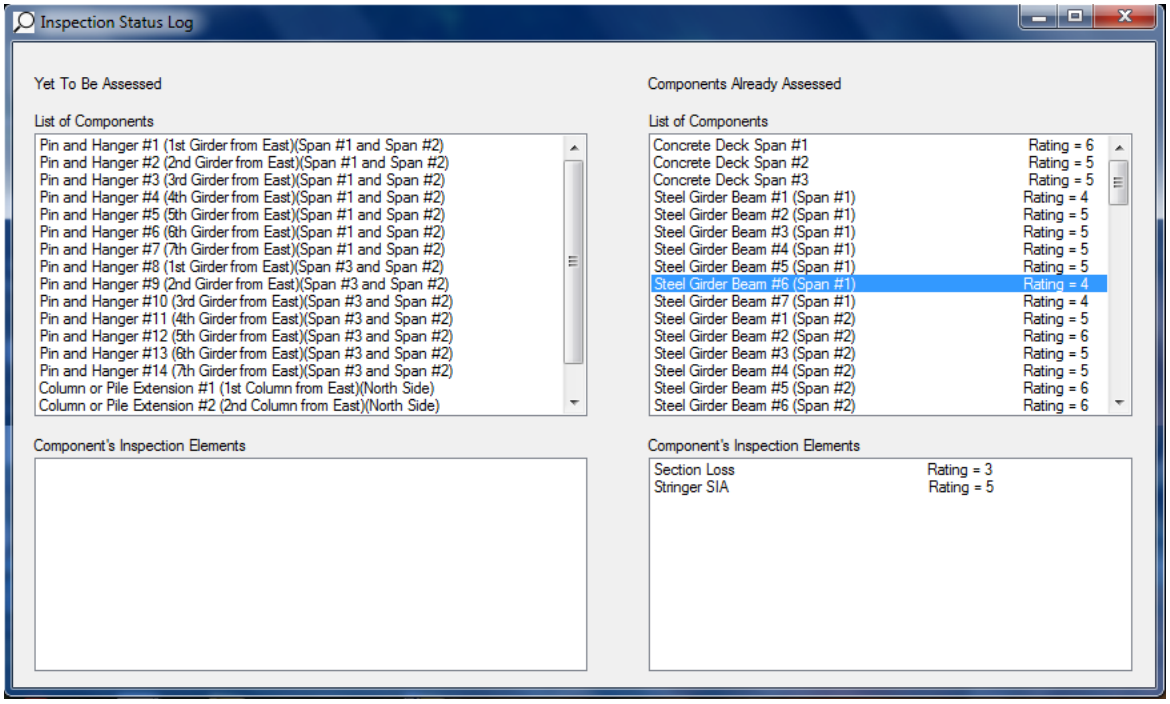
When all elements corresponding to the component are assigned ratings, the BIT computes the component's condition rating and suggests it to the inspector. The inspector can either accept the suggested component condition rating or refine it using functionality that supports condition assessment decision-making.

The BIT allows the inspector to track the progress of the inspection operation by keeping a log of those components and elements that have been assessed and those that are yet to be assessed. The inspection progress log can be accessed by clicking on the button corresponding to the log in the main inspection window. Figure 7.16 shows the inspection progress log captured during an inspection routine of the CSB.



Bridge: I-275 NB over Telegraph Road (US-24)
Inspection Date: 5/1/2013
Component: Expansion Joint (Strip-seal)
Inspector Name: Kurt Zolnierek
Agency/Consultant: MDOT
Suggested Component Rating: 7.00
For additional support: See SHM and Equivalent Element data in Main Window
Select Final Component Rating: 1 2 3 4 5 6 7 8 9
Select Element: Expansion Joint
Select Element Rating: 7
Comments: Joint 2s is a strip seal joint with no problems noted.

Figure 7.15: BIT GUI for uploading condition assessment information



Yet To Be Assessed

List of Components

Pin and Hanger #1 (1st Girder from East)(Span #1 and Span #2)
Pin and Hanger #2 (2nd Girder from East)(Span #1 and Span #2)
Pin and Hanger #3 (3rd Girder from East)(Span #1 and Span #2)
Pin and Hanger #4 (4th Girder from East)(Span #1 and Span #2)
Pin and Hanger #5 (5th Girder from East)(Span #1 and Span #2)
Pin and Hanger #6 (6th Girder from East)(Span #1 and Span #2)
Pin and Hanger #7 (7th Girder from East)(Span #1 and Span #2)
Pin and Hanger #8 (1st Girder from East)(Span #3 and Span #2)
Pin and Hanger #9 (2nd Girder from East)(Span #3 and Span #2)
Pin and Hanger #10 (3rd Girder from East)(Span #3 and Span #2)
Pin and Hanger #11 (4th Girder from East)(Span #3 and Span #2)
Pin and Hanger #12 (5th Girder from East)(Span #3 and Span #2)
Pin and Hanger #13 (6th Girder from East)(Span #3 and Span #2)
Pin and Hanger #14 (7th Girder from East)(Span #3 and Span #2)
Column or Pile Extension #1 (1st Column from East)(North Side)
Column or Pile Extension #2 (2nd Column from East)(North Side)

Component's Inspection Elements

Components Already Assessed

List of Components

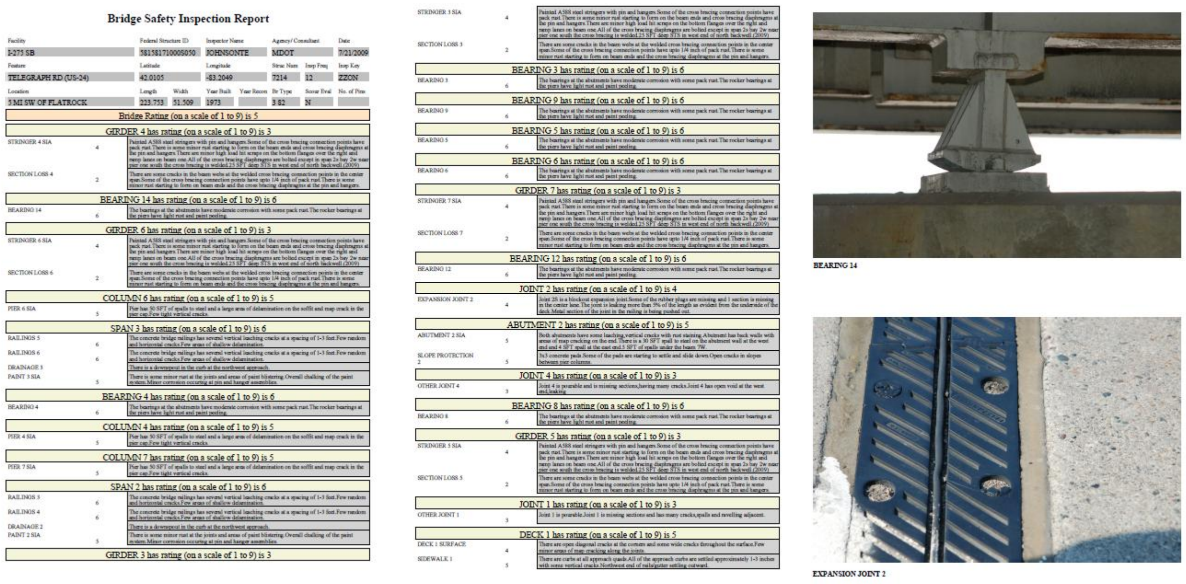
Concrete Deck Span #1	Rating = 6
Concrete Deck Span #2	Rating = 5
Concrete Deck Span #3	Rating = 5
Steel Girder Beam #1 (Span #1)	Rating = 4
Steel Girder Beam #2 (Span #1)	Rating = 5
Steel Girder Beam #3 (Span #1)	Rating = 5
Steel Girder Beam #4 (Span #1)	Rating = 5
Steel Girder Beam #5 (Span #1)	Rating = 5
Steel Girder Beam #6 (Span #1)	Rating = 4
Steel Girder Beam #7 (Span #1)	Rating = 4
Steel Girder Beam #1 (Span #2)	Rating = 5
Steel Girder Beam #2 (Span #2)	Rating = 6
Steel Girder Beam #3 (Span #2)	Rating = 5
Steel Girder Beam #4 (Span #2)	Rating = 5
Steel Girder Beam #5 (Span #2)	Rating = 6
Steel Girder Beam #6 (Span #2)	Rating = 6

Component's Inspection Elements

Section Loss	Rating = 3
Stringer SIA	Rating = 5

Figure 7.16: BIT inspection progress log captured during an inspection routine of the CSB

Upon completing an inspection, the inspector can generate information-rich inspection reports by clicking on the button corresponding to report generation in the main inspection window. The inspection report is customized to the format used by MDOT and is generated as a Portable Document Format (PDF) file. Figure 7.17 shows excerpts from the bridge inspection report of an inspection routine conducted on the CSB.



Bridge Safety Inspection Report

Facility	Federal Structure ID	Inspector Name	Agency/Consultant		Date
I-275 SB	58158171000S050	JOHNSONTE	MDOT		7/21/2009

Feature	Latitude	Longitude	Struc Num	Insp Freq	Insp Key
TELEGRAPH RD (US-24)	42.0105	-83.2049	7214	12	ZZON

Location	Length	Width	Year Built	Year Recon	Br Type	Scour Eval	No. of Pins
5 MI SW OF FLATROCK	223.753	51.509	1973		3 82	N	

Bridge Rating (on a scale of 1 to 9) is 5

GIRDER 4 has rating (on a scale of 1 to 9) is 3

STRINGER 4 SIA	4	Painted A588 steel stringers with pin and hangers.Some of the cross bracing connection points have pack rust.There is some minor rust starting to form on the beam ends and cross bracing diaphragms at the pin and hangers.There are minor high load hit scraps on the bottom flanges over the right and ramp lanes on beam one.All of the cross bracing diaphragms are bolted except in span 2s bay 2w near pier one south the cross bracing is welded.25 SFT deep STS in west end of north backwall (2009)
SECTION LOSS 4	2	There are some cracks in the beam webs at the welded cross bracing connection points in the center span.Some of the cross bracing connection points have upto 1/4 inch of pack rust.There is some minor rust starting to form on beam ends and the cross bracing diaphragms at the pin and hangers.

BEARING 14 has rating (on a scale of 1 to 9) is 6

BEARING 14	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

GIRDER 6 has rating (on a scale of 1 to 9) is 3

STRINGER 6 SIA	4	Painted A588 steel stringers with pin and hangers.Some of the cross bracing connection points have pack rust.There is some minor rust starting to form on the beam ends and cross bracing diaphragms at the pin and hangers.There are minor high load hit scraps on the bottom flanges over the right and ramp lanes on beam one.All of the cross bracing diaphragms are bolted except in span 2s bay 2w near pier one south the cross bracing is welded.25 SFT deep STS in west end of north backwall (2009)
SECTION LOSS 6	2	There are some cracks in the beam webs at the welded cross bracing connection points in the center span.Some of the cross bracing connection points have upto 1/4 inch of pack rust.There is some minor rust starting to form on beam ends and the cross bracing diaphragms at the pin and hangers.

COLUMN 6 has rating (on a scale of 1 to 9) is 5

PIER 6 SIA	5	Pier has 50 SFT of spalls to steel and a large area of delamination on the soffit and map crack in the pier cap.Few tight vertical cracks.

SPAN 3 has rating (on a scale of 1 to 9) is 6

RAILINGS 5	6	The concrete bridge railings has several vertical leaching cracks at a spacing of 1-3 feet.Few random and horizontal cracks.Few areas of shallow delamination.
RAILINGS 6	6	The concrete bridge railings has several vertical leaching cracks at a spacing of 1-3 feet.Few random and horizontal cracks.Few areas of shallow delamination.
DRAINAGE 3		There is a downspout in the curb at the northwest approach.
PAINT 3 SIA	5	There is some minor rust at the joints and areas of paint blistering.Overall chalking of the paint system.Minor corrosion occuring at pin and hanger assemblies.

BEARING 4 has rating (on a scale of 1 to 9) is 6

BEARING 4	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

COLUMN 4 has rating (on a scale of 1 to 9) is 5

PIER 4 SIA	5	Pier has 50 SFT of spalls to steel and a large area of delamination on the soffit and map crack in the pier cap.Few tight vertical cracks.

COLUMN 7 has rating (on a scale of 1 to 9) is 5

PIER 7 SIA	5	Pier has 50 SFT of spalls to steel and a large area of delamination on the soffit and map crack in the pier cap.Few tight vertical cracks.

SPAN 2 has rating (on a scale of 1 to 9) is 6

RAILINGS 3	6	The concrete bridge railings has several vertical leaching cracks at a spacing of 1-3 feet.Few random and horizontal cracks.Few areas of shallow delamination.
RAILINGS 4	6	The concrete bridge railings has several vertical leaching cracks at a spacing of 1-3 feet.Few random and horizontal cracks.Few areas of shallow delamination.
DRAINAGE 2		There is a downspout in the curb at the northwest approach.
PAINT 2 SIA	5	There is some minor rust at the joints and areas of paint blistering.Overall chalking of the paint system.Minor corrosion occuring at pin and hanger assemblies.

GIRDER 3 has rating (on a scale of 1 to 9) is 3

STRINGER 3 SIA	4	Painted A588 steel stringers with pin and hangers.Some of the cross bracing connection points have pack rust.There is some minor rust starting to form on the beam ends and cross bracing diaphragms at the pin and hangers.There are minor high load hit scraps on the bottom flanges over the right and ramp lanes on beam one.All of the cross bracing diaphragms are bolted except in span 2s bay 2w near pier one south the cross bracing is welded.25 SFT deep STS in west end of north backwall (2009)
SECTION LOSS 3	2	There are some cracks in the beam webs at the welded cross bracing connection points in the center span.Some of the cross bracing connection points have upto 1/4 inch of pack rust.There is some minor rust starting to form on beam ends and the cross bracing diaphragms at the pin and hangers.

BEARING 3 has rating (on a scale of 1 to 9) is 6

BEARING 3	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

BEARING 9 has rating (on a scale of 1 to 9) is 6

BEARING 9	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

BEARING 5 has rating (on a scale of 1 to 9) is 6

BEARING 5	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

BEARING 6 has rating (on a scale of 1 to 9) is 6

BEARING 6	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

GIRDER 7 has rating (on a scale of 1 to 9) is 3

STRINGER 7 SIA	4	Painted A588 steel stringers with pin and hangers.Some of the cross bracing connection points have pack rust.There is some minor rust starting to form on the beam ends and cross bracing diaphragms at the pin and hangers.There are minor high load hit scraps on the bottom flanges over the right and ramp lanes on beam one.All of the cross bracing diaphragms are bolted except in span 2s bay 2w near pier one south the cross bracing is welded.25 SFT deep STS in west end of north backwall (2009)
SECTION LOSS 7	2	There are some cracks in the beam webs at the welded cross bracing connection points in the center span.Some of the cross bracing connection points have upto 1/4 inch of pack rust.There is some minor rust starting to form on beam ends and the cross bracing diaphragms at the pin and hangers.

BEARING 12 has rating (on a scale of 1 to 9) is 6

BEARING 12	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

JOINT 2 has rating (on a scale of 1 to 9) is 4

EXPANSION JOINT 2	4	Joint 2S is a blockout expansion joint.Some of the rubber plugs are missing and 1 section is missing in the center lane.The joint is leaking more than 5% of the length as evident from the underside of the deck.Metal section of the joint in the railing is being pushed out.

ABUTMENT 2 has rating (on a scale of 1 to 9) is 5

ABUTMENT 2 SIA	5	Both abutments have some leaching,vertical cracks with rust staining.Abutment has back walls with areas of map cracking on the end.There is a 30 SFT spall to steel on the abutment wall at the west end and 4 SFT spall at the east end.5 SFT of spalls under the beam 7W.
SLOPE PROTECTION 2	5	3x3 concrete pads.Some of the pads are starting to settle and slide down.Open cracks in slopes between pier columns.

JOINT 4 has rating (on a scale of 1 to 9) is 3

OTHER JOINT 4	3	Joint 4 is pourable and is missing sections,having many cracks.Joint 4 has open void at the west end,leaking.

BEARING 8 has rating (on a scale of 1 to 9) is 6

BEARING 8	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

GIRDER 5 has rating (on a scale of 1 to 9) is 3

STRINGER 5 SIA	4	Painted A588 steel stringers with pin and hangers.Some of the cross bracing connection points have pack rust.There is some minor rust starting to form on the beam ends and cross bracing diaphragms at the pin and hangers.There are minor high load hit scraps on the bottom flanges over the right and ramp lanes on beam one.All of the cross bracing diaphragms are bolted except in span 2s bay 2w near pier one south the cross bracing is welded.25 SFT deep STS in west end of north backwall (2009)
SECTION LOSS 5	2	There are some cracks in the beam webs at the welded cross bracing connection points in the center span.Some of the cross bracing connection points have upto 1/4 inch of pack rust.There is some minor rust starting to form on beam ends and the cross bracing diaphragms at the pin and hangers.

JOINT 1 has rating (on a scale of 1 to 9) is 3

OTHER JOINT 1	3	Joint 1 is pourable.Joint 1 is missing sections and has many cracks,spalls and ravelling adjacent.

DECK 1 has rating (on a scale of 1 to 9) is 5

DECK 1 SURFACE	4	There are open diagonal cracks at the corners and some wide cracks throughout the surface.Few minor areas of map cracking along the joints.
SIDEWALK 1	5	There are curbs at all approach quads.All of the approach curbs are settled approximately 1-3 inches with some vertical cracks.Northwest end of rail/gutter settling outward.

BEARING 14

EXPANSION JOINT 2

Figure 7.17: Bridge inspection report of the CSB generated by the BIT

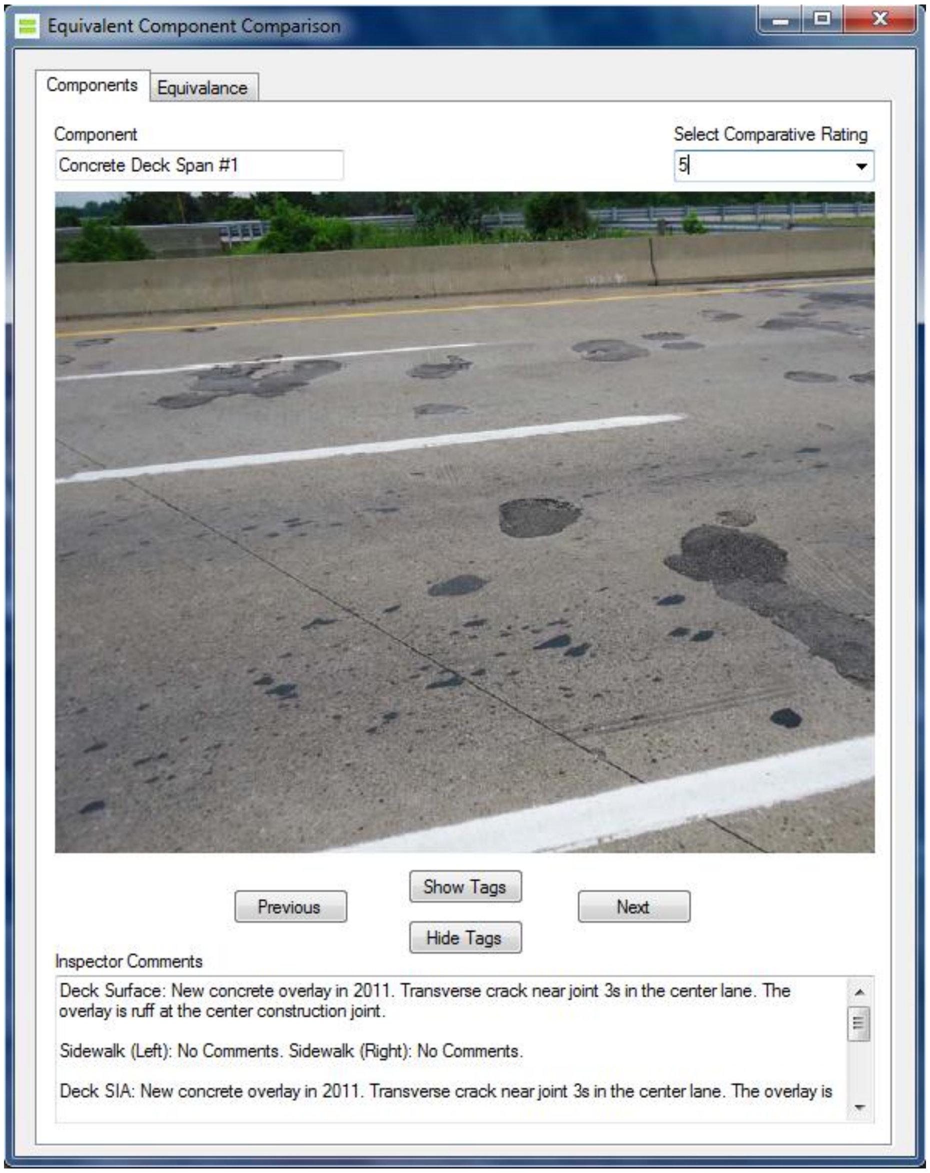
7.3.2.2 Decision-Making Support for Condition Assessment

As mentioned previously, inspectors assess the condition of the bridge's primary components and rate them using the NBIS 0 to 9 rating scale (MDOT 2011, Farrar 2008). The NBIS rating scale, summarized in Table 7.1, results in subjective ratings with significant variability in assigned condition ratings (Moore et al 2001). The BIT improves the objectivity of assigned condition ratings by providing the inspector with condition assessment decision-making support functionality. Condition assessment decision-making support is provided using two methods: 1) equivalent component comparison, and 2) SHM analysis.

Rating	Observed Condition
9	Excellent Condition – like new.
8	Very Good Condition – no problems noted.
7	Good Condition – some minor problems observed.
6	Satisfactory Condition – structural components show minor deterioration.
5	Fair Condition – all primary structural components are sound but may have minor corrosion, cracking, or chipping. May include minor erosion on bridge piers.
4	Poor Condition – advanced corrosion, deterioration, cracking, or chipping. Also significant erosion of concrete bridge piers.
3	Serious Condition – corrosion, deterioration, cracking and chipping, or erosion of concrete bridge piers. Seriously affected deck, superstructure, or substructure. Local failures are possible.
2	Critical Condition – advanced deterioration of deck, superstructure, or substructure components. May have cracks in steel or concrete, or erosion may have removed substructure support. It may be necessary to close the bridge until corrective action is taken.
1	Imminent Failure Condition – major deterioration or corrosion in deck, superstructure, or substructure components. Obvious vertical or horizontal movement affecting structure stability. Bridge is closed to traffic but corrective action may put it back in light service.
0	Failed Condition – out of service, beyond corrective action.
N	Not applicable.

Table 7.1: NBIS 0 to 9 condition rating scale

Equivalent component comparison functionality allows the inspector to draw on the experience and judgment of peers by comparing the condition of the component in context with equivalent components across the rating scale. Upon query, the inspector is directed to the form, shown in Figure 7.18, which provides the functionality required for comparing equivalent components.



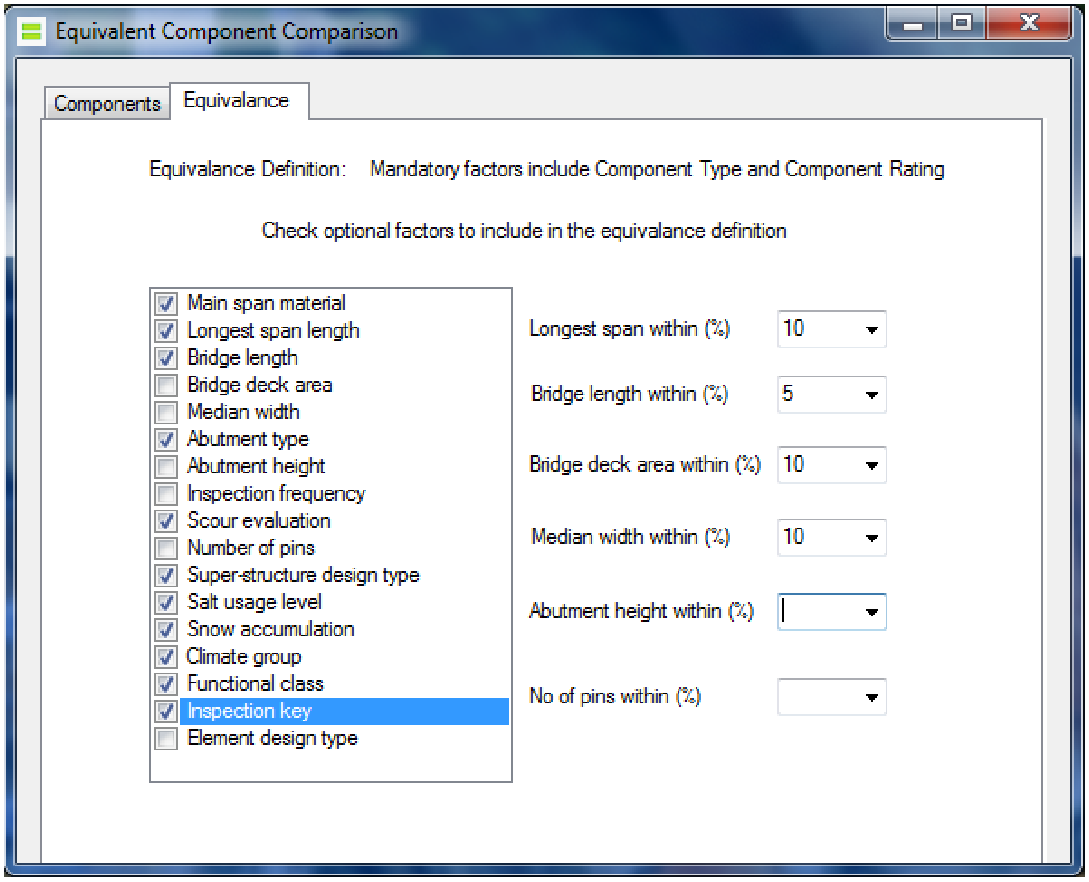
Deck Surface: New concrete overlay in 2011. Transverse crack near joint 3s in the center lane. The overlay is ruff at the center construction joint.

Sidewalk (Left): No Comments. Sidewalk (Right): No Comments.

Deck SIA: New concrete overlay in 2011. Transverse crack near joint 3s in the center lane. The overlay is

Figure 7.18: BIT functionality to compare the condition of equivalent components assigned similar rating

The inspector selects the condition rating deemed appropriate for the component in context using the dropdown box. Upon selection, the BIT queries the CDR to retrieve assessment information for similar components belonging to equivalent bridges with the same condition rating. The inspector views captured multimedia, inspection notes, and observations associated with similarly rated equivalent components in the inventory. The inspector refines the component's condition rating by browsing for condition concurrence among equivalent components. The BIT allows the inspector to define similarity among bridges based on bridge characteristics and traffic volume, as shown in Figure 7.19.



Equivalance Definition: Mandatory factors include Component Type and Component Rating

Check optional factors to include in the equivalance definition

- [x] Main span material
- [x] Longest span length
- [x] Bridge length
- [ ] Bridge deck area
- [ ] Median width
- [x] Abutment type
- [ ] Abutment height
- [ ] Inspection frequency
- [x] Scour evaluation
- [ ] Number of pins
- [x] Super-structure design type
- [x] Salt usage level
- [x] Snow accumulation
- [x] Climate group
- [x] Functional class
- [x] Inspection key
- [ ] Element design type

Longest span within (%)	10
Bridge length within (%)	5
Bridge deck area within (%)	10
Median width within (%)	10
Abutment height within (%)	
No of pins within (%)	

Figure 7.19: BIT functionality to define similarity among bridges to search for equivalent components in the inventory

The second method employed by the BIT to provide condition assessment decision-making support is based on SHM analysis. Upon query, the inspector is directed to the web-based sensor analytics form, as shown in Figure 7.20. The URL of the web-form is customized based on the component being assessed. The web-form has URL links to raw and processed SHM information corresponding to the component in context. The inspector requests the desired analysis by clicking on the corresponding URL to invoke appropriate Python scripts in the sensor analytics client. The sensor analytics client runs the SHM algorithms and returns the resulting Google Chart object as an output. The interactive Google Chart object is embedded into the web-browser and presented to the inspector. The inspector then interprets the presented SHM analysis and refines the component's condition rating.

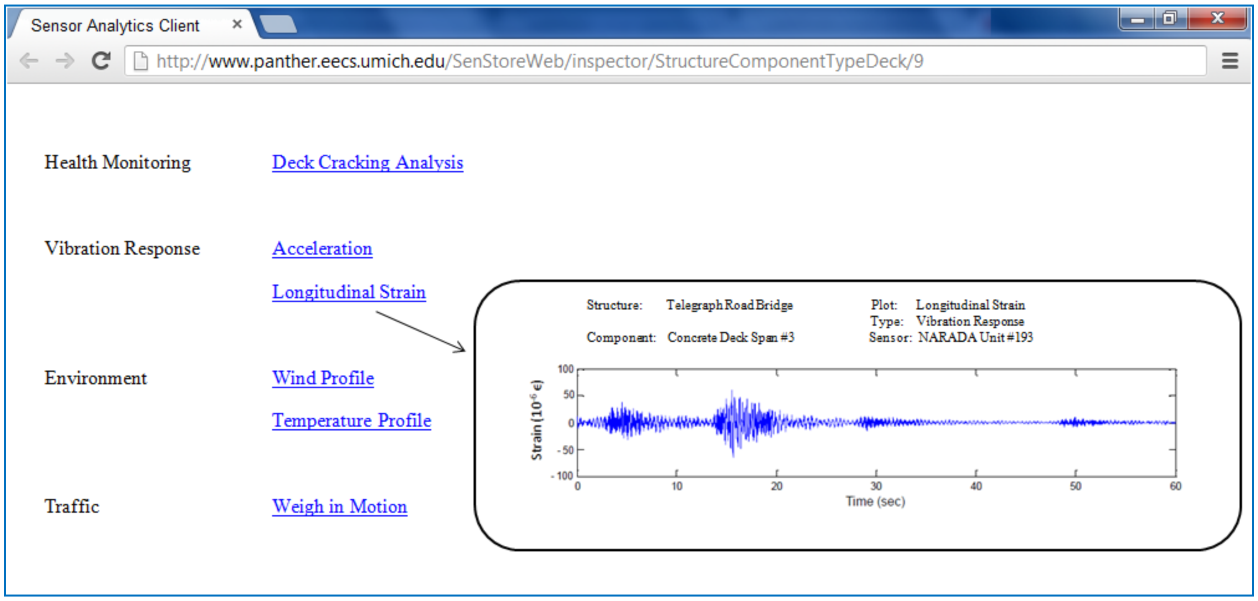


Figure 7.20: SHM analysis functionality available in the BIT

The raw SHM information available to the inspector contains customized time-histories of sensed parameters including accelerometer, strain gauge, and temperature data. The processed SHM information available to the inspector is directly dependent on the SHM algorithms implemented in the sensor analytics client corresponding to the component in context. As a proof of concept, Section 7.3.3 describes specific examples of SHM analysis that can be intelligently interpreted by the inspector to refine component condition rating. The four examples described correspond to the deck (cracking analysis), expansion joint (lock-up analysis), pin and hanger (fatigue analysis), and deck-girder system (composite action) components of highway bridges.

7.3.3 Structural Health Monitoring Analysis

7.3.3.1 Deck Cracking Analysis

Bridge deck cracking has historically been an issue for bridge owners due to cracking's role in reducing a bridge's structural life. The degradation of concrete decks is accelerated significantly by water, de-icing salts, and other chemicals that permeate through cracks in bridge decks thus reducing structural life and increasing maintenance needs (Alampalli 2001). Vibration severity has been identified as the most significant factor influencing bridge deck cracking with longer spans exhibiting more deck cracking than shorter spans. Traffic volume was found to have a low significance compared to vibration severity and deck span length whereas bridge bearing was found to have no influence on deck cracking severity (Alampalli 2001).

The vibration severity of bridge decks is measured using strain gauge sensors that monitor the vibration response of the structure. A typical vibration response of a highway bridge deck is plotted, in Figure 7.21, as a time-history graph of the longitudinal strain observed in the bridge deck. The deck strain is measured for a defined period of time (4 hours), the peak absolute dynamic strain measured in that period of time is recorded in the CDR, and the process is continuously repeated.

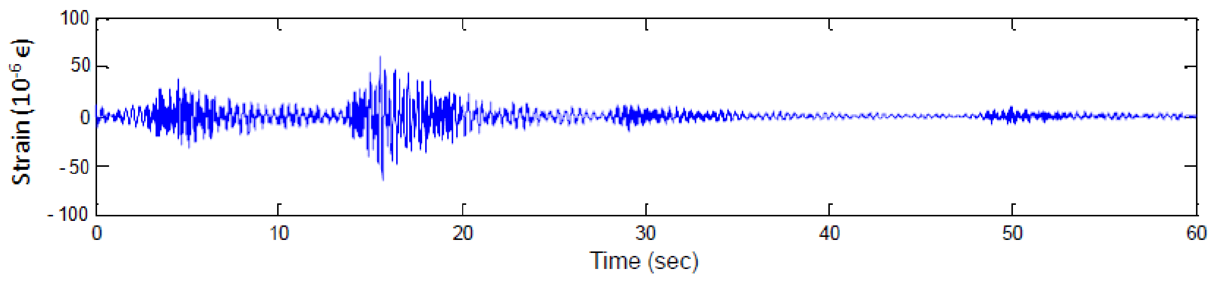


Figure 7.21: Typical vibration response of a highway bridge deck subject to traffic loads

The cumulative response of the deck is presented to the inspector by plotting the peak absolute dynamic strain histogram, as shown in Figure 7.22. The sensor analytics plot compares the histogram corresponding to the current inspection (plotted between the last recorded inspection and the current inspection in progress) (shown in red) and the last recorded inspection (plotted between second-to-last recorded inspection and the last recorded inspection) (shown in blue).

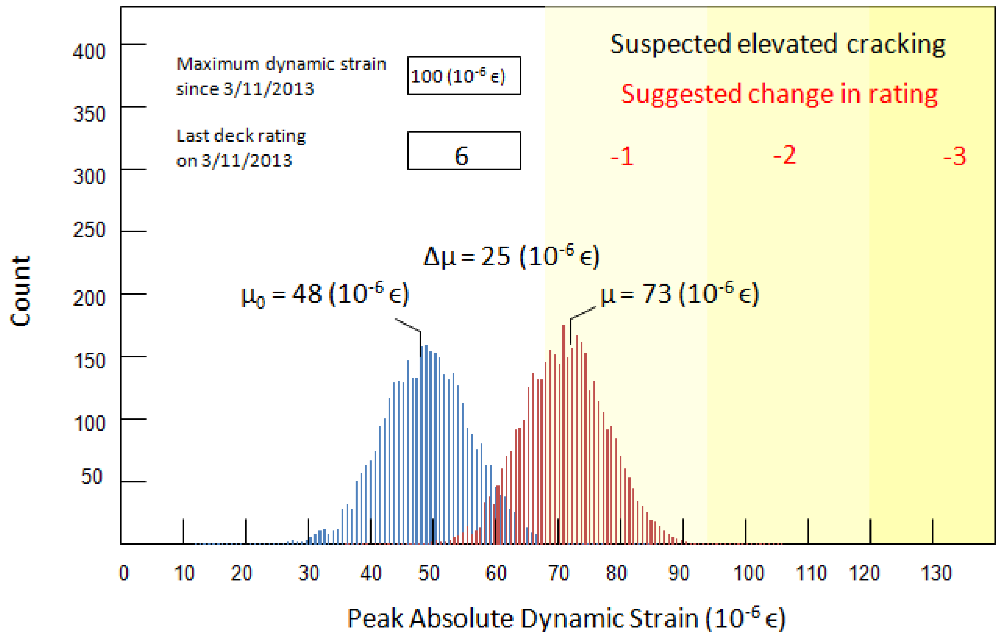


Figure 7.22: Peak absolute dynamic strain histogram for deck cracking analysis

The average peak absolute dynamic strain recorded for the current inspection (μ) is usually higher than that recorded for the preceding inspection (μ_0) due to the combined effect of deck cracking and fatigue. The difference in the average peak absolute dynamic strain ($\Delta\mu$) between current and preceding inspections is used to quantify the deterioration in deck condition and to suggest appropriate changes to the condition rating of the deck component.

7.3.3.2 Pin and Hanger Fatigue Analysis

Pin and hanger assemblies, shown in Figure 7.23, are structural components designed to allow expansion movement and rotation. Pin and hanger joints are usually found in multi-span bridges built prior to 1970. Although pin and hanger designs are no longer used, many bridges with these assemblies are in service and will remain so in the future. Additionally, these structural components will experience aging and age-related deterioration in the next few decades. Thus, it is important to pay special attention to the condition assessment and maintenance of these assemblies during inspections (Ryan et al 2006).

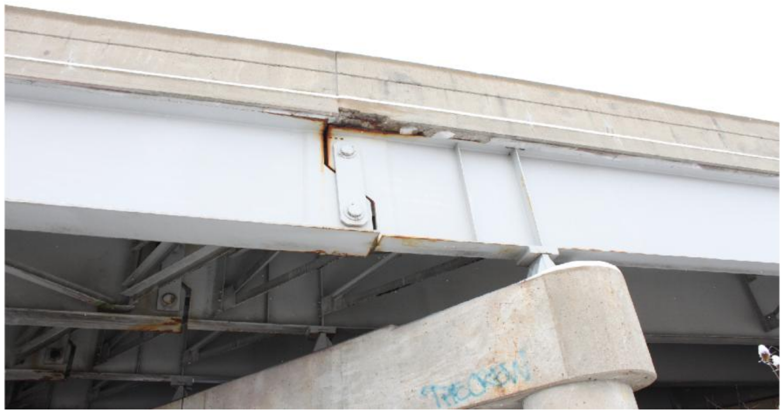


Figure 7.23: Pin and hanger assembly in the CSB

Common deteriorations observed on pin and hanger assemblies include paint failures, corrosion, collision damage, overloads, heat damage, and fatigue cracking. While paint failures, corrosion, and collision damage can be visually observed by the inspector, the BIT can prove to be very useful in helping the inspector assess overloads, heat damage, and fatigue cracking by presenting relevant sensor analysis.

ASTM International defines the fatigue life of a structural unit as the number of stress cycles of a specified character that a specimen sustains before failure of a specific nature occurs (Stephens and Fuchs 2001). The fatigue life of a pin and hanger assembly can be equated to a certain number of stress cycles of particular magnitude before the assemblies experiences fatigue failure. The major variables that influence the stress magnitude in a pin and hanger assembly are truck weights, the lateral distribution of truck loads, impact factor, span lengths, beam redundancy, the angle of crossing and corrosion in the pin and hanger. Additional variables that influence fatigue life are AADTT, the number of traffic lanes, the percentage of trucks in a given lane, and the number of stress occurrences per truck (Bershing 2001). By applying material properties, the stress magnitudes experienced by the pin and hanger components are directly reflected in the strain experienced by the pin and hanger link plate.

Strain gauge sensors mounted on the pin and hanger assembly component are used to measure the time-history of the strain experienced by the pin and hanger assembly for a defined period of time (4 hours). The time-history of the strain is composed of a series of strain cycles with varying amplitudes. The number of cycles corresponding to each of the observed strain amplitudes is calculated and stored in the CDR along with the time-stamps noting the observation period. This process is repeated continuously. The fatigue accumulated in the pin and hanger is presented to the inspector by plotting the strain amplitude histogram, as shown in Figure 7.24. Figure 7.24 compares the strain amplitude histogram recorded since monitoring began (shown in gray) as well as the histogram recorded since the last inspection (shown in red) for a typical pin and hanger assembly.

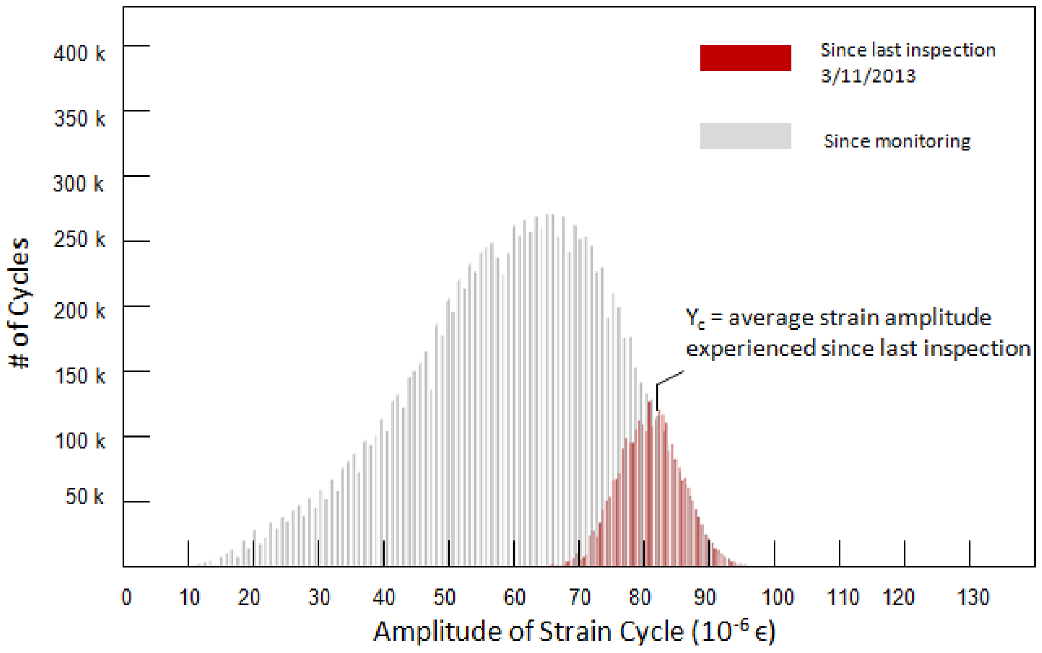


Figure 7.24: Strain cycle amplitude histogram for pin and hanger fatigue analysis

The fatigue life rate is plotted versus time, as shown in Figure 7.25, and presented to the inspector. The fatigue life rate (ΔY) is estimated as the difference in the average strain amplitude recorded since the last inspection (Y_c) and the average strain amplitude recorded between the last inspection and the second-to-last inspection (Y_{c-1}). The change in average strain amplitude is expected to increase for every successive inspection due to the loss in outstanding fatigue life. As long as the change in the average strain amplitude remains within reasonable limits (shown in green), the component is expected to last for its entire design life. Significant condition deterioration due to fatigue is observed as a rapid change in average strain amplitudes recorded during successive measurement periods (shown in yellow and red). The trend in average strain amplitudes is used to quantify the fatigue deterioration in pin and hanger assemblies, and to suggest appropriate changes to the condition rating.

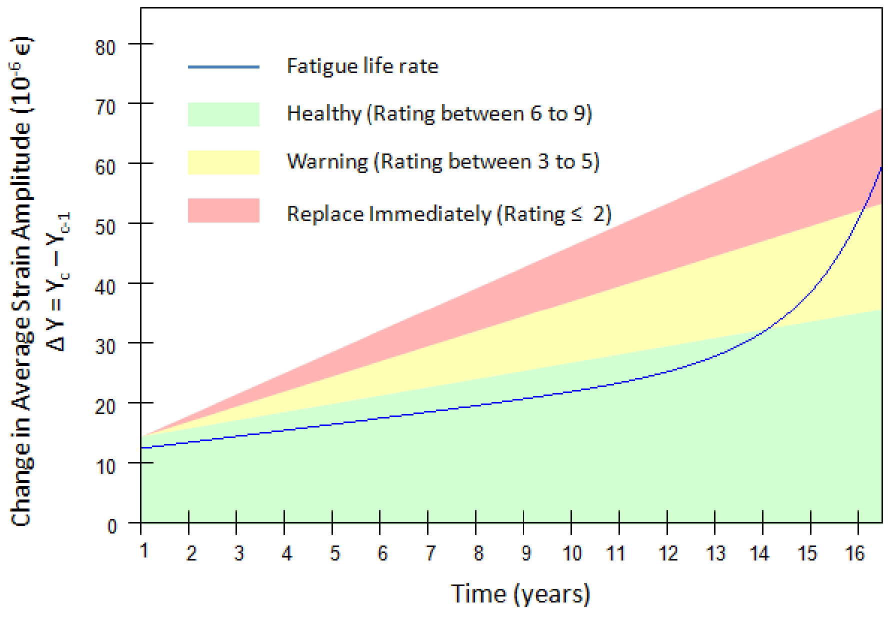


Figure 7.25: Fatigue life rate vs time plot for pin and hanger fatigue analysis

7.3.3.3 Deck-Girder Composite Action Analysis

Typical highway bridge superstructures are built as a composite construction of concrete deck and steel girders. The composite deck-girder system of the CSB is shown in Figure 7.26. The deck and girder are physically connected using bolts, adhesive, and shear connections to behave as a single composite unit when subject to loading. The deck and girders are positively connected, preventing any slip between the two, thus allowing the deck slab to contribute to the girder strength. The composite action of the deck-girder system results in significant improvements in bending strength and stiffness of the bridge, allowing the use of much lighter girders than those used in non-composite girder systems for the same span and loading (Brown et al 2003).

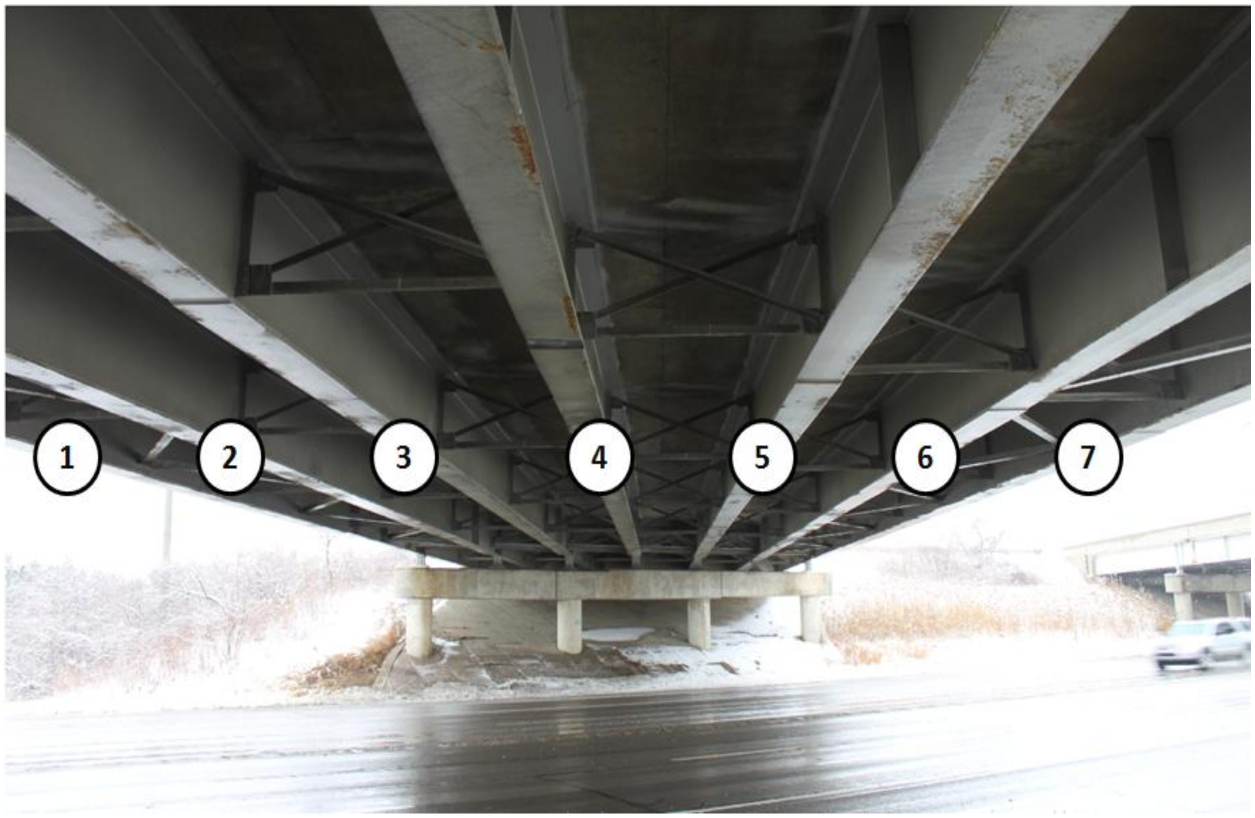


Figure 7.26: Underside of the CSB deck and the 7 girders that form the composite action system

The effectiveness of composite action is measured by comparing the designed neutral axis with the observed neutral axis of the deck-girder system. The location of the design neutral axis is determined for capacity loading and is represented using a probability distribution. Strain gauge sensors are attached to the bottom and top flanges of the 7 girders in the CSB to record strain history. Using strains measured at the bottom flange since the last scheduled inspection, and the time-correspondent strains at the top flange, the location of the neutral axis is estimated as a distribution (Brown et al 2003). The sensor analytics client presents the inspector with a visualization of the distribution of the designed (shown in red) and observed (shown in blue) neutral axis as shown in Figure 7.27. The sensor analytics client also estimates the safety factor (β) by calculating the margin between the capacity (designed) and demand (observed) neutral axis distribution.

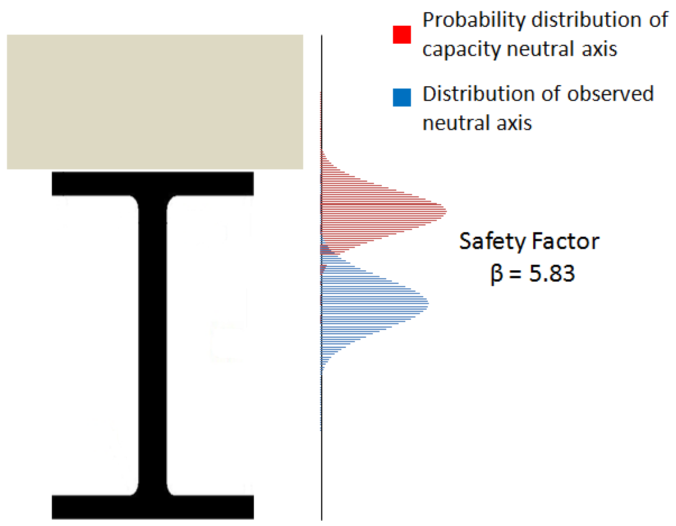

Probability distribution of capacity neutral axis
Distribution of observed neutral axis
Safety Factor
β = 5.83

Figure 7.27: Location of the designed (capacity) and observed (demand) neutral axis of the deck-girder system exhibiting composite action

The inspector assesses the condition of the girder system based on the estimated safety factors. As a rule of thumb, girder systems in acceptable condition have large safety factors (greater than 8), whereas those in critical condition have small safety factors (less than 2). A safety factor that is estimated between acceptable and critical conditions is considered to be a warning indicating inefficient load transfer between the deck and girder components due to deteriorating connections. The inspector thus objectively assesses the condition of the deck-girder composite system and recommends corrective action as deemed necessary.

7.3.3.4 Expansion Joint Lock-Up Analysis

Multi-span bridges are characterized by gaps between deck spans to accommodate any movement of the deck. Deck movement is primarily induced by thermal expansion and contraction caused by temperature changes. Secondary causes for deck movement include dynamic loading, foundation settlement, creep, shrinkage, sway caused by wind, and seismic events (Jones 2011). In bridge construction, an expansion joint, shown in Figure 7.28, is a mid-structure assembly designed to hold deck components together while safely absorbing the stress on construction material caused by deck movements. Expansion joints support the surfacing, or provide a running surface across the expansion gap allowing changes to the size of the expansion gap without damage; expansion joints also usually prevent the passage of water below deck level (Barnard and Cuninghame 1997).

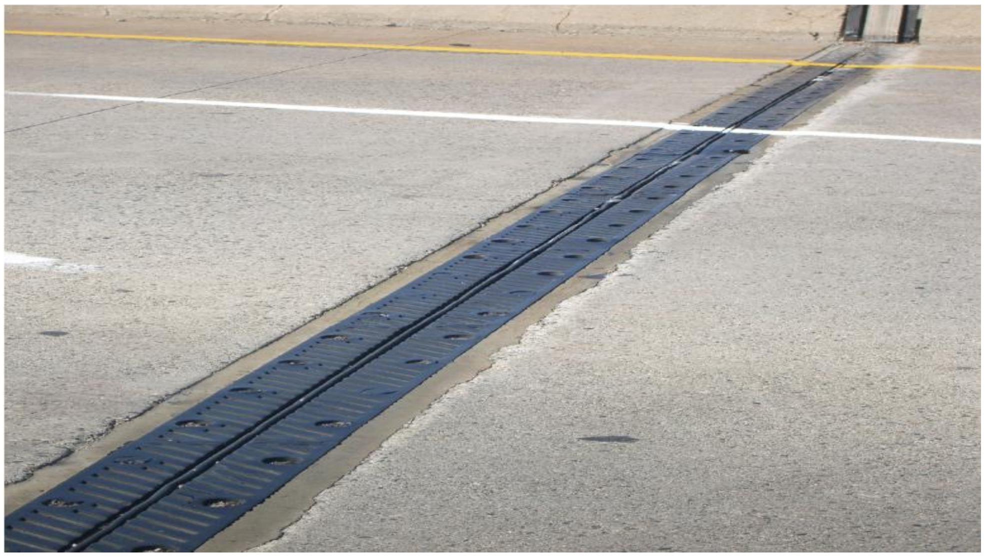


Figure 7.28: Strip seal expansion joint in the CSB

The expansion joint performance is measured using strain gauges that record the movement of the decks connected by the expansion joint. The expansion joint performance is measured as the time-history of strain experienced by the joint for a defined period of time (4 hours). The time-history of the temperature is measured using temperature sensors in the pervasive sensor network of the bridge. The average strain and the temperature for the defined time period are estimated and stored as a single strain-temperature data point in the CDR. The aforementioned process of estimating and storing strain-temperature data points is repeated continuously.

The relationship between average strain and temperature (thermal expansion curve), during a time period, is determined by plotting the best fit curve for the strain-temperature data points recorded in the same time period. The inspector is presented with the thermal expansion curves, shown in Figure 7.29, corresponding to two periods of time: 1) the time between the last inspection and the current inspection (shown in red), and 2) the time between the beginning of monitoring and the last inspection (shown in blue).

Expansion joints are designed to expand (or contract) linearly with an increase (or decrease) in temperature to prevent decks from cracking and buckling. The observed thermal expansion best fit curve resembles a linear relationship for expansion joints with good condition. A non-linear manifestation of the observed thermal expansion curve indicates deterioration in the expansion joint condition. Inefficient expansion joint performance is induced due to debris collection in the seals, thermal cycles, traffic cycles, and poor workmanship at installation. The inspector refines the expansion joint condition rating by estimating the deviation of the thermal expansion curve, observed since the last inspection, from the idealized linear thermal expansion curve.

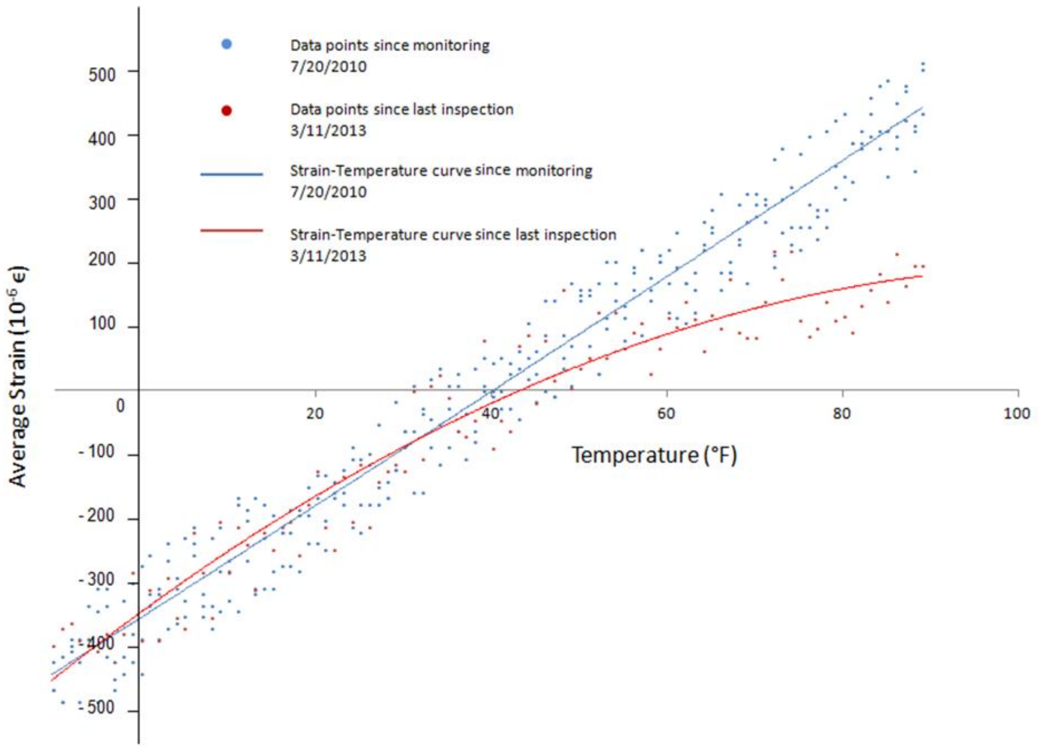


Figure 7.29: Thermal expansion curves for expansion joint lock-up

7.4 Limitations and Challenges in Adopting the BIT

Factors that limit and challenge the adoption of the BIT developed in this research can be classified into five main categories: bridge environment, context identification, data comprehensiveness, health monitoring algorithms, and run-time complexity.

Bridge environment: The bridge inspection tools developed in this chapter are limited by the fact that their implementation was tested using the CSB. While GPS localization and head orientation tracking technologies are independent of any installed infrastructure, UWB localization technology—used to track the inspector in GPS-denied environment—requires infrastructure to be installed in the underside of the bridge. Thus, the operability of CONOID is restricted to those bridges in the inventory where UWB-based localization infrastructure is installed. Additionally, the BIT communicates with the CDR and the sensor analytics client through the internet using a 3G mobile phone network. Thus, the toolkits presented in this chapter may experience deteriorated performance during inspection operations assessing bridges located in regions where 3G wireless internet network coverage is poor or non-existent.

Context identification: The BIT identifies the component of interest to the inspector by either manual selection or automated selection using CONOID. CONOID predicts the inspector's component of interest by constructing and interpreting the inspector's spatial-context based on parameters sensed by the employed tracking technology. The inspector's location is tracked by using GPS where available, and UWB-based localization technologies for regions where GPS is unavailable (for example, the underside of the bridge), respectively. The inspector's head orientation is tracked by using a magnetic compass or non-magnetic orientation trackers. The accuracy of CONOID is limited by the uncertainty in the employed tracking technologies as documented in Chapter 6.

Data comprehensiveness: In order for bridge inspectors to maximize the benefits of the BIT functionality, it is imperative for the CDR to have a comprehensive record of the entire bridge inventory within their jurisdiction, their corresponding inspection history, and SHM data. Bridge inventory and corresponding inspection history data typically exists with the owner organizations in their paper form and/or in a proprietary database, and must be transferred to the CDR through appropriate methods. To collect SHM information, bridge inventory needs to be installed with appropriate sensor networks and corresponding SHM tables must be configured appropriately to store the collected data. A substantial and diverse bridge inventory repository serves to accentuate the power of the equivalent component comparison and the historic inspection functionality of the BIT.

Structural health monitoring algorithms: SHM support provided to the inspectors is limited by several factors. SHM for bridge infrastructure is limited to certain types of damages that can be sensed in components. SHM research is focused on developing algorithms and methods to estimate appropriate component condition variables based on SHM sensor data. However, methods to convert these condition variables into bridge condition assessment ratings are currently limited.

Run-time complexity: Finally, the adoption of the bridge inspection solution discussed in this chapter is limited by the run-time complexity of the BIT, CONOID, and sensor analytics client. The run-time complexity of CONOID is independent of the size of the bridge inventory as CONOID downloads the component geometry information relevant to the bridge being assessed and stores it locally before the inspection begins. The locally stored component geometry information is used as an input to CONOID's containment and prioritization algorithm, thus making it real-time, as demonstrated in Chapter 6. The BIT functionalities and queries—as current implemented—do not consider the size of the database. As the CDR currently contains inventory, inspection, and SHM information corresponding only to the CSB, the run-time of the BIT functionality and the sensor analytics client is in the sub-second range. However, the run-time performance of the BIT functionality is expected to deteriorate for larger inventory sizes. Thus, ensuring that the BIT functionalities operate in near real-time—for large inventories—is a key aspect in adopting the bridge inspection solutions developed in this research.

7.5 Summary and Conclusions

A smoothly functioning transportation system is vital to the economic interest of a country as well as the safety of the travelling public (FHWA 2013). Highway bridge infrastructure is a key aspect of the U.S. transportation system. Efficient maintenance is important for preserving the integrity and improving the condition of an aging national bridge infrastructure system. Highway bridge condition is typically assessed by professional bridge inspectors using visual inspections and accompanying non-destructive field tests (Farrar 2008; FHWA 2011; Moore 2011). Recently, the use of widespread deployment of low-cost SHM sensor networks has been proposed for the large-scale monitoring of bridge condition (Kurata et al 2011a; Kurata et al 2011b; O'Connor et al 2012; Swartz et al 2007).

This chapter presents a context-aware bridge inspection solution that leverages SHM data and the cumulative condition assessment knowledge of bridge inspectors. The developed solution utilizes the CDR to store information describing bridge infrastructure inventory, historic condition assessment, and SHM sensor readings. The chapter presents the BIT—a new software solution that provides condition assessment decision-making support to bridge inspection documentation. The BIT is seamlessly integrated with the CONOID application to provide the option of utilizing context-aware software navigation. The BIT allows inspectors access to condition assessment data corresponding to equivalent components, as recorded by other responders, thus allowing them to draw on the collective experience and judgment of peers. Condition assessment decision-making support is also provided by integrating the BIT with functionality that allows inspectors access to intelligent interpretation of SHM data.

The research presented in this chapter allows bridge component condition assessment variables—acquired through SHM algorithms and methods—to be co-related with time-correspondent component condition rating, assessed by inspectors, thus providing better insight into SHM for structural engineers. An enhanced understanding of the relationship between SHM and condition ratings allows the development of more comprehensive SHM algorithms to support condition assessment decision-making and establishes baseline data for existing infrastructure in the case of future structural events.

The implementation of the bridge inspection solution developed in this research introduces a greater degree of objectivity into condition ratings assessed by bridge inspectors. The condition assessment data gathered using the presented methods helps in effectively rating bridge infrastructure capacity and allocating maintenance resources (Alampalli and Ettouney 2010). The effective maintenance of bridge condition allows for the maximization of commerce traffic flow, the safe utilization of transportation infrastructure, and increased public confidence in bridge infrastructure.

7.6 Acknowledgments

The author would like to gratefully acknowledge the support offered by the U.S. Department of Commerce, and the National Institute of Standards and Technology (NIST) Technology

Innovation Program (TIP) under Cooperative Agreement Number 70NANB9H9008. The author would like to specially acknowledge the efforts of Prof. Jerome P. Lynch and his PhD students Yilan Zhang and Sean O'Connor for providing valuable insights into the field of bridge infrastructure health monitoring. The author would also like to thank Atul Sandur, Prof. Atul Prakash, Dr. Sharada Alampalli, and Dr. Gwendolyn van der Linden for their assistance in implementing some of the research presented in this chapter.

7.7 References

Abudayyeh, O., Attanayake, U., Abdel-Qader, I., Aktan, H., Almaita, E., and Barbera, J. (2011). "An Information System for a Sensor-Based Bridge Health Monitoring System," Proc. of the 5th Int. Conference on Information Technology (ICIT 2011), Amman, Jordan.

Adams, T.M., Juni, E., Siddiqui, M.K., and Dzienkowski, J.E. (2005). "Integrated Field and Office Tools for Bridge Management," Transportation Research Record, Transportation Research Board, Vol. 1933(1), pp. 35–43.

Alampalli, S. (2001). "Correlation between Bridge Vibration and Bridge Deck Cracking: A Qualitative Study," Special Report No. 136, Transportation Research and Development Bureau, New York State Department of Transportation, Albany, NY, pp. 1–19.

Alampalli, S., and Ettouney, M. (2010). "Automated Decision Making Tool for Bridge Managers," Proc. of the 2010 Conference on Nondestructive Evaluation (NDE)/Nondestructive

Testing (NDT) for Highways and Bridges: Structural Materials Technology (SMT), American Society of Nondestructive Testing (ASNT), New York, NY.

Atasoy, G., Tang, P., Zhang, J., and Akinci, B. (2010). "Visualizing Laser Scanner Data for Bridge Inspection," The 27th Int. Symp. on Automation and Robotics in Construction (ISARC 2010), Bratislava, Slovakia.

Barnard, C.P., and Cuninghame, J.R. (1997). Practical Guide to the Use of Bridge Expansion Joints, Transportation Research Laboratory, Crowthorne, Berkshire, England.

Bershing, S. (2001). "Michigan's Link Plate and Pin Assemblies," Construction and Technology Research Record Issue No. 92, Construction and Technology Division, Michigan Department of Transportation (MDOT), Lansing, MI, pp. 1–4.

BridgeInspect™ (2012). "BridgeInspect Software for Bridge Inspectors and Managers" – Website of BridgeInspect™ <http://www.bridgeinspect.com/> as accessed on 15th December, 2012.

BridgeWorks.NET (2012). "Washington State Department of Transportation's Bridge Inspection and Reporting Software" – Website of the Washington State Department of Transportation <http://www.wsdot.wa.gov/LocalPrograms/Bridge/BridgeWorks.htm> as accessed on 15th December, 2012.

Brown, M.C., Gomez, J.P., Cousins, T.E., and Barton, F.W. (2003). "Composite Action in a Steel Girder Span with Precast Deck Panels: The I-81 Bridge Over the New River in Radford, Virginia," Report No. FHWA/VTRC 04-R4, Virginia Transportation Research Council (VTRC) in cooperation with the U.S. Department of Transportation's Federal Highway Administration (FHWA), Charlottesville, VA, pp. 1–17.

Casas, J., and Cruz, P. (2003). "Fiber Optic Sensors for Bridge Monitoring," Journal of Bridge Engineering, American Society of Civil Engineers (ASCE), Vol. 8(6), pp. 362–373.

Dally, J., and Riley, W. (2005). Experimental Stress Analysis, 4th Edition, College House Enterprises LLC, Knoxville, TN, ISBN: 0-9762413-0-7.

Dong, Y., and Song, R. (2010). "Bridges Structural Health Monitoring and Deterioration Detection – Synthesis of Knowledge and Technology," AUTC Report No. 309036, Alaska University Transportation Center (AUTC) and the Department of Civil and Environmental Engineering, University of Alaska Fairbanks, Fairbanks, AK.

Doornink, J.D., Phares, B.M., Zhou, Z., Ou, J., and Graver, T.W. (2004). "Fiber Bragg Grating Sensing for Structural Health Monitoring of Civil Structures," Int. Symp. on Advances and Trends in Fiber Optics and Applications (ATFO2004), Chongqing and Wuhan, China, Vol. 11(2), pp. 41–44.

Ettouney, M., and Alampalli, S. (2007). “Toolchest for Decision Making and SHM,” Proc. of the 16th Annual ASNT Research Symposium, American Society of Nondestructive Testing (ASNT), Orlando, FL.

Farrar, M. M. (2008). “Inspection,” The AASHTO Manual for Bridge Evaluation, 2nd Edition, with 2011 Interim Revisions, Section 4, American Association of State Highway and Transportation Officials, Washington, D.C., ISBN: 1-56051-496-1.

FHWA (2008). “More about AASHTO Bridge Management” – Website of the United States Department of Transportation’s Federal Highway Administration (FHWA)’s Asset Management Program < http://www.fhwa.dot.gov/infrastructure/asstmgmt/pontmore.cfm> dated 14th October, 2008, as accessed on 20th April, 2013.

FHWA (2011). “Structure Type by Year Built: Count of Bridges 2011” – Website of the United States Department of Transportation’s Federal Highway Administration (FHWA) <http://www.fhwa.dot.gov/bridge/nbi/yrblt11.cfm> dated 31st December, 2011, as accessed on 20th April, 2013.

FHWA (2013). “Highway Finance Data Collection – Our Nation’s Highways: 2010” – Website of the United States Federal Highway Administration (FHWA)’s Office of Highway Policy Information <http://www.fhwa.dot.gov/policyinformation/pubs/hf/pl10023/> as accessed on 20th April, 2013.

Flintsch, W.G., and Bryant Jr., J.W. (2006). “Asset Management Data Collection for Supporting Decision Processes,” <http://www.fhwa.dot.gov/asset/dataintegration/if08018/amdc_00.cfm>, Federal Highway Administration (FHWA), U.S. Department of Transportation, Washington, D.C.

Hao, S. (2009). "I-35W Bridge Collapse," Journal of Bridge Engineering, American Society of Civil Engineers (ASCE), Vol. 15(5), pp. 608–614.

Hearn, P. (2008). “MSU Invention Excites Concrete—in the Interest of Bridge Safety, that is” – Website of the Mississippi State University of Agriculture and Applied Science <http://www.msstate.edu/web/media/detail.php?id=2986> dated 28th July, 2005, as accessed on 20th April, 2013.

Housner, G.W., Bergman, L.A., Caughey, T.K., Chassiakoa, A.G., Claus, R.O., Masri, S.F., Skelton, R.E., Soong, T.T., Spencer, B.F., and Yao, J.T.P. (1997). “Structural Control: Past, Present, and Future,” ASCE Journal of Engineering Mechanics, American Society of Civil Engineers (ASCE), Vol. 123(9), pp. 897–971.

Inaudi, D. and Vurpillot, S. (1999). “Monitoring of Concrete Bridges with Long-Gage Fiber Optic Sensors,” Journal of Intelligent Material Systems and Structures, Vol. 10(4), pp. 280–292.

Indiana State BIAS (2012). "Indiana Bridge Inspection Application System" – Website of the Indiana State Bridge Inspection Application System <https://inbridges.indot.in.gov/> as accessed on 15th December, 2012.

Jones, P. (2011). "Inspection Guidance for Bridge Expansion Joints: Part 1 – Reference Guide," <http://www.bridgeforum.org/bof/meetings/bof34/>, Transport for London, Greater London Authority, London, England, pp. 1–49.

Karbhari, V.M. (2005). "Chapter 5 – Health Monitoring, Damage Prognosis and Service-Life Prediction – Issues Related to Implementation," Sensing Issues in Civil Structural Health Monitoring, Edited by Ansari, F., Springer, ISBN: 1-4020-3660-4, pp. 301–310.

Koch, C., and Brilakis, I. (2011). "Pothole Detection in Asphalt Pavement Images," Advanced Engineering Informatics, Elsevier Science, New York, NY, Vol. 25(3), pp. 507–515.

Kurata, M., Kim, J., Zhang, Y., Lynch, J.P., van der Linden, G.W., Jacob, V., Thometz, E., Hipley, P., and Sheng, L.H. (2011a). "Long-Term Assessment of an Autonomous Wireless Structural Health Monitoring System at the New Carquinez Suspension Bridge," SPIE's 18th Annual Int. Symp. on Smart Structures and Materials + Nondestructive Evaluation and Health Monitoring, International Society for Optics and Photonics (SPIE), San Diego, CA, pp. 798312–798322.

Kurata, M., Lynch, J.P., van der Linden, G.W., Hipley, P., and Sheng, L.H. (2011b). "Application of an Automated Wireless Structural Monitoring System for Long-Span Suspension Bridges," AIP Conference Proc., American Institute of Physics (AIP), Vol. 1335, pp. 33–40.

Li, Y.F., Wang, J., Wang, M.Z., Deluccia, J., and Laird, C. (1999). "Development of the Electrochemical Fatigue Sensor for Evaluating Fatigue Damage," Proc. of the TMS Fall Meeting '99, The Minerals, Metals, and Materials Society (TMS), Cincinnati, OH, pp. 333–339.

MDOT (2011). "Highway Bridge Report – October 1, 2011", Michigan Department of Transportation (MDOT), Lansing, MI. pp. 1–2.

Mercado, E.J., and Woodroof, J.R. (2008). "The Pneumatic Scour Detection System: Development and Case History," Proc. of the 2008 Conference on Nondestructive Evaluation (NDE)/Nondestructive Testing (NDT) for Highways and Bridges: Structural Materials Technology (SMT), American Society of Nondestructive Testing (ASNT), Oakland, CA.

Miyamoto, A. (2008). "IT-based Bridge Health Monitoring," Proc. of the 23rd Int. Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC 2008), Shimonoseki City, Japan, pp. SP-9–SP-14.

Moore, M., Phares, B., Graybeal, B., Rolander, D. and Washer, G. (2001). "Reliability of Visual Inspection for Highway Bridges, Volume I, Final Report," FHWA Final Report No. FHWA-RD-01-020, Federal Highway Administration (FHWA), Washington, D.C.

Moore, W.H. (2011). “Table 1-28: Condition of U.S. Highway Bridges,” National Transportation Statistics, Bureau of Transportation Statistics, Washington, D.C.

Mufti, A. (2001). “Guidelines for Structural Health Monitoring,” ISIS Canada Design Manual No. 2, Intelligent Sensing for Innovative Structures (ISIS) Canada Resource Center, University of Manitoba, Winnipeg, Manitoba.

NTSB (2008). “Collapse of the I-35W Highway Bridge, Minneapolis, Minnesota, August 1, 2007,” Highway Accident Report NTSB/HAR-08-03, National Transportation Safety Board (NTSB), Washington, D.C., pp. 1–162.

O’Connor, S., Lynch, J.P., Ettouney, M., van der Linden, G., and Alampalli, S. (2012). “Cyber-enabled Decision Making System for Bridge Management using Wireless Monitoring Systems: Telegraph Road Bridge Demonstration Project,” Proc. of the 2012 Conference on Nondestructive Evaluation (NDE)/Nondestructive Testing (NDT) for Highways and Bridges: Structural Materials Technology (SMT), American Society of Nondestructive Testing (ASNT), New York, NY.

Phares, B., Wipf, T., Greimann, L., and Lee, Y. (2005). “Health Monitoring of Bridge Structures and Components Using Smart Structure Technology,” Wisconsin Highway Research Program Report No. WHRP 05-03, Wisconsin Department of Transportation (WisDOT), Madison, WI, Vol. 1, pp. 1–53.

Reis, R. and Gallaher, M. (2006). “Evaluation of the VTI ECI-1 Embedded Corrosion Instrument,” Caltrans Report No. FHWA/CA/TL-2003/07/ECI-1, Materials Engineering and Testing Services, California Department of Transportation (Caltrans), Sacramento, CA.

Roberts, G.W., Meng, X., Meo, M., Dodson, A., Cosser, E., Iuliano, E., and Morris, A. (2003). “A Remote Bridge Health Monitoring System using Computational Simulation and GPS Sensor Data,” Proc. of the 11th FIG Symposium on Deformation Measurements, International Federation of Surveyors (FIG), Santorini, Greece.

Roberts, G.W., Meng, X., and Dodson, A. (2004). “Integrating a Global Positioning System and Accelerometers to Monitor the Deflection of Bridges,” Journal of Surveying Engineering, American Society of Civil Engineers (ASCE), pp. 65–72.

Ryan, T.W., Raymond, Hartle, R.A., Mann, J.E., and Danovich, L.J. (2006). “Bridge Inspector’s Reference Manual,” FHWA Publication No. FHWA NHI 03-001, Federal Highway Administration (FHWA), United States Department of Transportation, Washington, D.C.

Stephens, R.I., and Fuchs, H.O. (2001). Metal Fatigue in Engineering, 2nd Edition, John Wiley and Sons, New York, NY, ISBN: 0-471-51059-9, pp. 69–70.

Swartz, R.A., Zimmerman, A., and Lynch, J.P. (2007). "Structural Health Monitoring System with the Latest Information Technologies," Proc. of 5th Infrastructure and Environmental Management Symp., Yamaguchi, Japan, pp. 1–28.

Tang, P., and Akinci, B. (2012). "Formalization of Workflows for Extracting Bridge Surveying Goals from Laser-Scanned Data," Automation in Construction, Elsevier Science, New York, NY, Vol. 22(3), pp. 306–319.

van der Linden, G.W., Emami-Naeini, A., Zhang, Y., and Lynch, J.P. (2013). "Cyber-Infrastructure Design and Implementation for Structural Health Monitoring," SPIE's 20th Annual Int. Symp. on Smart Structures and Materials + Nondestructive Evaluation and Health Monitoring, International Society for Optics and Photonics (SPIE), San Diego, CA, pp. 869419–869428.

Wardhana, K., and Hadipriono, F.C. (2003). "Analysis of Recent Bridge Failures in the United States," Journal of Performance of Constructed Facilities, American Society of Civil Engineers (ASCE), Vol. 17(3), pp. 144–150.

Yang, L. (2008). Techniques for Corrosion Monitoring, Southwestern Research Institute (San Antonio, TX), Woodhead Publishing, Cambridge, England.

Chapter 8

Conclusion

8.1 Significance of the Research

The workspace environment in certain industries—such as manufacturing—is strictly controlled and remains relatively unchanged over time. Thus, domain operations and processes are defined using standardized workspace parameters and are largely independent of the need to sense and react to changes in the environment. However, the civil infrastructure industry is characterized by mobile human user and machine agents actively engaged in real-time decision-making tasks in a dynamic, unstructured, and evolving workspace environment (Son et al 2008). Field personnel spend a significant amount of time in manually sensing an evolving workspace environment, analyzing large volumes of acquired information, making mission-critical decisions in real-time, and communicating streamlined information with one another.

The effort and time spent in manually analyzing the workspace environment and accessing the relevant information required to support important decision-making tasks leads to a considerable loss in productivity and money in the civil infrastructure industry (Aziz et al 2005; Fathi et al 2006; Zou et al 2006). Context-aware information retrieval provides timely and accurate access to streamlined project information by monitoring physical and environmental context (Khoury 2009). Context-awareness does not simply allow computing devices to sense and react to changes in the environment; its intrinsic worth is established in its potential to support critical decision-making tasks on behalf of human and machine agents.

Prior efforts in developing context-aware capabilities in the civil infrastructure industry have typically focused on designing wearable computing systems (Korutem et al 1999; Pascoe et al 1998), developing the architecture for mobile distributed context-aware computing systems (Aziz et al 2006; Behzedan et al 2008; Khoury and Kamat 2009), and tracking resource locations (Omar and Ballal 2009; Skibniewski and Jang 2006; Teizer et al 2008). Recent efforts have also involved the introduction of a human user's head orientation as a spatial-context parameter, thus allowing context-aware applications to define a user's spatial-context with much greater precision than is possible with location alone (Khoury 2009).

The research documented in this dissertation presents a framework for designing context-aware computing applications to support real-time decision-making in the civil infrastructure domain. The framework addresses three specific challenges underlying the implementation of context-aware applications: 1) ubiquitous localization, 2) interpreting contextual objects, and 3) data management and organization. This has led to the development of a ubiquitous localization system that integrates infrastructure-based localization technology with inertial navigation. The framework abstracts the spatial-context of human user and machine agents by constructing representative geometric models of the field of view and the region of influence, respectively. Proximate selection methodologies used to identify, prioritize, and interpret objects of spatial interest are also presented. The dissertation also discusses the suitability of adopting standard product models and databases to organize and manage data objects in context-aware computing applications.

Additionally, this research has also implemented several applications—based on the context-aware computing framework—to support real-life decision-making tasks in the civil infrastructure industry. These applications not only serve to illustrate the usefulness of the context-aware computing framework, but also have significant social and economic implications. For example, the controlled drilling application warns drilling personnel when a drill bit tip is about to strike rebar or utility lines, thus helping preserve the structural integrity of concrete decks and preventing utility strike accidents, and so reducing losses of life, capital, and opportunity. The automated fault detection for Heating, Ventilation, and Air Conditioning (HVAC) distribution systems leverages contextual parameters—such as HVAC system performance and repair activity status—to generate a plan of action for the field inspectors and repair personnel to follow, thus eliminating the time and effort spent by personnel to manually do the same based on their judgment and experience. The context-aware bridge inspection allows inspectors to access streamlined information that supports condition assessment decision-making, thus introducing objectivity to visual condition assessment by providing concurrence with the sensed Structural Health Monitoring (SHM) data.

Figure 8.1 summarizes these applications based on their real-time standards, spatial-context accuracy, and non-spatial-context descriptiveness. The controlled drilling application is subject to hard real-time standards and relies on highly precise spatial-context construction. However, it does not utilize non-spatial contextual parameters to construct the context of the drill. The automated fault detection application for HVAC systems is subject to soft real-time standards and relies only on non-spatial contextual parameters. The context-aware bridge inspection solution is also subject to soft real-time standards. However, it relies on both spatial and non-spatial contextual parameters to construct the inspector's context.

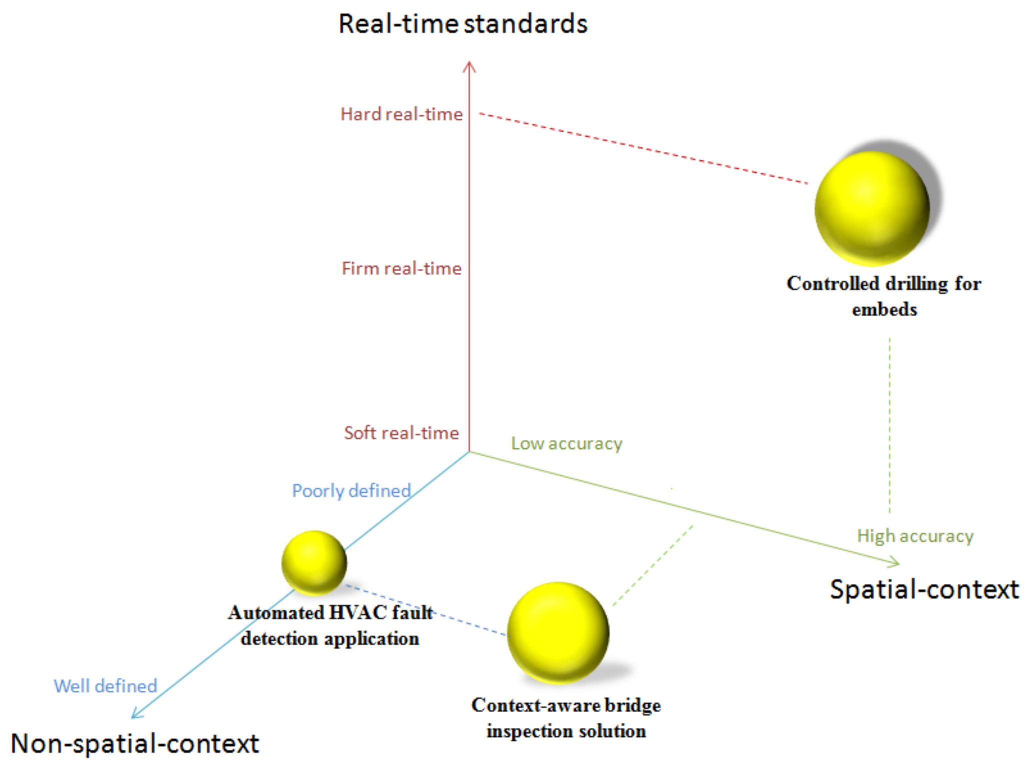


Figure 8.1: Taxonomic summary of the context-aware applications presented in this dissertation

8.2 Research Contribution

This research contributes to the civil infrastructure industry by improving job safety and productivity with the implementation of context-aware applications developed using the proposed framework. For construction contractors, engineers, and maintenance personnel, this will transform traditional means of accessing, sorting, and streamlining project information, and thus improve their efficiencies and decision-making capabilities. This research also contributes to the maintenance of bridge infrastructure fleets by improving the accuracy in condition assessment methods, and thus increasing the efficiency of allocating maintenance resources to address deterioration.

The individual research challenges that were successfully investigated and overcome in the preceding chapters are summarized in the following bullet points:

- A general-purpose conceptual framework for developing real-time context-aware applications in the civil infrastructure industry.
- A ubiquitous localization algorithm based on the integration of infrastructure-based localization systems with inertial navigation technology. The algorithm was used to implement three versions of the Integrated Tracking System (ITS) for ubiquitous localization of human users in dynamic, unstructured environments.
- A collision warning system that controls drilling for embeds into reinforced concrete bridge decks, and warns drilling personnel when they are about to strike buried rebar or utility lines.
- An automated fault detection system for localizing faults in a HVAC distribution network and for generating a plan of action for field inspectors and repair personnel.
- An understanding of how errors in location and head orientation tracking technologies affect the accuracy of identifying the object of interest to a mobile highway bridge inspector.

- A bridge inspection database that stores bridge inventory, component geometry, condition assessment information, and the documentation supporting assessment decisions.
- A context-aware computing application known as Contextual Object Identifier (CONOID) that identifies those bridge components that are of interest to the inspector.
- Innovative bridge inspection software known as Bridge Inspection Toolkit (BIT) that provides inspectors with decision-making support for condition assessment documentation.

Chapter 2: This chapter first discusses various types of real-time systems based on the characteristics of their performance-time deadlines. The main contribution of this chapter is the framework developed in this research for designing context-aware applications to support real-time decision-making tasks in the civil infrastructure domain. The chapter then discusses the components of the proposed framework and the key drivers involved in creating such applications. The chapter presents four essential categories of contextual parameters—identity, time, activity status, and spatial information—that are used by the framework to define the fully qualified context of end human user and machine agents. The chapter then discusses data models that are most suitable to define and store information about the physical and functional characteristics of operational facilities, and to serve as data repositories for context-aware computing applications. Methods to construct an agent's context are then discussed with a special emphasis on the abstraction of the spatial context of mobile human user and machine agents by constructing representative geometric models of the field of view and the region of influence, respectively. The chapter finally discusses methods to interpret the context, make decisions, and communicate these decisions to the end user agents.

Chapter 3: This chapter covers the overarching architecture and algorithm of the ITS for integrating infrastructure-based localization systems and infrastructure-independent localization technologies for the ubiquitous tracking of a mobile human user. The chapter then covers the development of a version of the ITS based on integrating Real-Time Kinematic Global Positioning System (RTK-GPS) and Personal Dead Reckoning (PDR), as well as several validation experiments that were conducted for varying complexities of the mobile user's trajectory proving the sustainability of the ITS. Next, the chapter presents the development and implementation of a version of the ITS based on integrating a database of pre-determined known correction point locations and PDR. As a further improvement, a hybrid tracking system based on PDR and human intelligence was also developed and validated. Finally, the chapter discusses the advantages and disadvantages of the developed ubiquitous localization technology and compares them to infrastructure-based technologies and inertial navigation–based localization.

Chapter 4: The primary contribution of this chapter is the development of a real-time context-aware monitoring system that warns drilling personnel when they are about to strike buried rebar or utility lines while drilling for embeds into reinforced concrete decks. The chapter first presents the motivating scenario of drilling for embeds into reinforced concrete bridge decks where monitoring systems could have a significant impact in avoiding striking concealed rebar and utility infrastructure. The chapter then proposes two potential methods to address the problem of monitoring the process of drilling for embeds in real-time. The two methods—the drill feedback control approach and the laser projector-based guidance approach—were then implemented and validated in a test bed using a rebar cage designed as a mockup of a railway bridge rebar cage. The methods presented in this chapter could significantly improve production for the concrete deck placement operation, avoiding the time and cost needed to place and remove dowels; the method could also shorten the project duration (Saidi et al 2011).

Chapter 5: The primary contribution of this chapter is the development of a BIM-based automated fault detection system for HVAC distribution networks. The chapter investigates the process of inspecting and repairing damaged HVAC equipment as a motivating scenario for developing BIM-based decision-making applications. The chapter presents methods to model HVAC distribution networks and the deterioration in HVAC performance. A fault detection algorithm (Leckie and Dale 1997) that uses the minimum message length principle to localize those HVAC components most likely at fault is then presented. The chapter then presents a workflow method that generates the facility inspector's plan of action based on the adopted fault detection algorithm, and contextual parameters such as the sensed HVAC system performance and repair status. By dynamically generating the plan of action, the fault detection system guides the operation of HVAC repair and eliminates the time and effort spent by FM personnel to manually do the same based on their judgment and experience.

Chapter 6: This chapter presents a motivating scenario of conducting bridge inspection routines where integrating context-aware computing with new and existing bridge inspection reporting applications can potentially save the time spent in manually navigating the software, and can reduce the effort spent by inspectors to learn, remember and recall the taxonomic hierarchy and ontology of bridge components across a large fleet of bridges. The primary contribution of this chapter is a computing framework to automatically identify bridge components in the inspector's spatial context. The chapter also covers the containment and prioritization algorithms used by the framework to predict the component of interest to the inspector. The chapter then presents run-time and spatial complexity analyses of the framework. The chapter also discusses a sensitivity analysis study that characterized the framework's performance in predicting the component of interest based on simulations and field experiments. Finally, the chapter presents a workflow design for integrating context-awareness into bridge inspection documentation software. Supported by high-precision tracking technologies, the framework automatically identifies the component of interest, thus reducing the time and effort required to navigate the software.

Chapter 7: The primary contribution of this chapter is a context-aware bridge inspection solution that leverages SHM data and the cumulative condition assessment knowledge of bridge inspectors to provide decision-making support for condition assessment. The developed solution utilizes a database known as the Central Data Repository (CDR) to store bridge inspection data and SHM sensor readings. The chapter discusses the design of the CDR database from the perspective of documenting bridge infrastructure inventory, bridge component geometry, condition assessment data, and the documentation to support condition assessment decisions. The chapter then presents the BIT—a new software solution that provides condition assessment decision-making support to bridge inspection documentation. The BIT is seamlessly integrated with the CONOID application to provide the option of utilizing context-aware software navigation. The BIT allows inspectors access to condition assessment data corresponding to equivalent components, as recorded by other responders, thus allowing them to draw on the collective experience and judgment of peers. Condition assessment decision-making support is also provided by integrating the BIT with functionality that allows inspectors access to an intelligent interpretation of SHM data. The implementation of the bridge inspection solution developed in this chapter introduces a greater degree of objectivity into condition ratings assessed by bridge inspectors, thus improving the accuracy in estimating bridge infrastructure capacity and allocating maintenance resources (Alampalli and Ettouney 2010).

8.3 Future Directions of Research

This research has proposed a context-aware computing framework and has developed several applications, based on the proposed framework, to support real-time decision-making tasks in the civil infrastructure industry. However, the developed applications are subject to certain limitations that translate into interesting future research challenges. Some of these limitations and challenges have been mentioned in the conclusion section of each individual chapter. In addition, the following sections discuss specific thrusts that are viewed as logical directions for future research.

8.3.1 Productivity Analysis of Drilling Operations

In Chapter 4, the author presented a collision monitoring system that warns drilling personnel when they are about to strike buried rebar or utility lines by leveraging 3D imaging methods that map locations that are safe (or unsafe) for drilling. The specific strategy for conducting the productivity analysis is to compare the time for the manual method where the dowels are prepared, attached to the planks, and removed after concrete placement (traditional drilling method) versus mapping the rebar locations using either laser scanning or photogrammetry (controlled drilling method).

A Discrete Event Simulation (DES) approach has been proposed to calculate the potential time savings achieved from using either 3D imaging method (Saidi et al 2011). The DES model logic is derived from visits to the jobsite and discussions with engineers, managers, and foremen on projects involving the construction of reinforced concrete decks. For the 3D imaging methods, the tasks and durations for 3D imaging processes replace the tasks and durations corresponding to the handling of the dowels used in the traditional method. The DES models provide breakeven analysis for the use of 3D imaging methods in the construction of typical reinforced concrete deck spans where drilling for embeds is anticipated.

8.3.2 Exploring Alternative Tracking Technologies for Constructing Spatial-Context

In Chapter 6, the dissertation author discussed a sensitivity analysis study that characterized the uncertainty in constructing the spatial-context of a mobile human user due to uncertainties in the employed tracking technologies. It follows from the results of the sensitivity analysis that the uncertainty in constructing the spatial-context is significant when the user's head orientation is tracked using a magnetic compass. The magnetic compass relies on the earth's magnetic field to calculate the orientation readings, and thus becomes unreliable in the field as the magnetic fields of the bridge deck rebar, structural steel, and traffic interfere with the earth's magnetic field (Akula et al 2013).

An interesting challenge is to explore the feasibility of using high-accuracy gyroscopes and accelerometers to track the head orientation of the user. Since these technologies don't rely on the earth's magnetic field to calculate the orientation readings, they are not subject to errors due to dynamic interference with the same. However, these technologies may be subject to errors in tracking due to the drift accumulated during the operation. Also, the framework developed in Chapter 6 assumes that the mobile user's line of sight is in line with the head orientation; the framework does not consider eye motion. An interesting challenge would be to pursue the implementation of methods to track the user's line of sight by tracking the user's eyeball motion. This is commonly accomplished through computer vision techniques imaging pupil and corneal reflection of infrared light. Another possible venue of future research is the exploration of using computer vision techniques—including feature recognition and natural markers—to identify the objects of interest to the user.

8.3.3 Characterizing the Effects of Introducing Context-Awareness to Bridge Inspections

In Chapter 7, the author presented a context-aware bridge inspection methodology to improve the objectivity of condition assessment ratings by providing condition assessment decision-making support. Decision-making support is provided using context-aware computing, which results in a change in the inspector's workload dynamics. The effect on the workload dynamics can be evaluated through the NASA task load index (Hart and Staveland 1988), as shown in Table 8.1.

Criteria	Assessment Factors	Methodology	Assessment Metrics
Mental Demand	How mentally demanding was the task?	NASA Task Load Index Guidelines	Very Low to Very High
Physical Demand	How physically demanding was the task?	NASA Task Load Index Guidelines	Very Low to Very High
Temporal Demand	How hurried/rushed was the pace of the task?	NASA Task Load Index Guidelines	Very Low to Very High
Performance	How successful were you in accomplishing the task?	NASA Task Load Index Guidelines	Very Low to Very High
Effort	How hard did you have to work to accomplish the level of performance?	NASA Task Load Index Guidelines	Very Low to Very High
Frustration	How insecure, discouraged, irritated, stressed, or annoyed were you during the task?	NASA Task Load Index Guidelines	Very Low to Very High

Table 8.1: NASA task load index matrix for workload evaluation in human-machine systems

The introduction of context-awareness also results in a change in the technical aspects of bridge inspection documentation including the time and effort required to conduct the inspection, the richness of the documented information, the confidence in the assessment, and the accuracy of the assessed condition ratings. The effect on the technical factors can be evaluated using the matrix presented in Table 8.2.

Criteria	Assessment Factors	Methodology
Preparation	Time and effort taken to prepare and organize data before an inspection	MDOT interviews and statistics for traditional inspections; Zero time and effort for context-aware inspections
Field Work	Time and effort taken to conduct an inspection	Field experiment comparisons
Consolidation	Time and effort taken to consolidate and organize data before an inspection	MDOT interviews and statistics for traditional inspections; Zero time and effort for context-aware inspections
Information Richness	Media used for communication; Information customization	Information richness theory
Assessment Confidence	Inspector's confidence in ratings with decision-making support vs. traditional routine	Inspector community's self-assessment of confidence in ratings
Rating Accuracy	Improvement in rating accuracy with decision-making support	Does not currently exist

Table 8.2: Technical factors evaluation matrix for bridge inspection routines

To evaluate the effects of adopting the proposed inspection method, workload and technical factors can be evaluated in a two-step approach: first while performing inspection operations without any context-aware decision-making support (traditional method), and second while performing similar activity with decision-making support (proposed method). The proposed inspection method eliminates the time and effort associated with preparation and consolidation while documenting richer information to support condition assessment decisions. Methods to evaluate the accuracy of condition assessment ratings do not currently exist, and so pose an interesting challenge for future research.

8.4 References

Akula, M., Sandur, A., Kamat, V.R., and Prakash, A. (2013). “Context-Aware Framework for Highway Bridge Inspections,” Journal of Computing in Civil Engineering, American Society of Civil Engineers (ASCE), Reston, VA (In Press).

Alampalli, S., and Ettouney, M. (2010). "Automated Decision Making Tool for Bridge Managers," Proc. of the 2010 Conference on Nondestructive Evaluation (NDE)/Nondestructive Testing (NDT) for Highways and Bridges: Structural Materials Technology (SMT), American Society of Nondestructive Testing (ASNT), New York, NY.

Aziz, Z., Anumba, C.J., Ruikar, D., Carrillo, P.M., and Bouchlaghem, D.N. (2005). “Context Aware Information Delivery for on-Site Construction Operations,” 22nd CIBW78 Conference on Information Technology in Construction, Institute for Construction Informatics, Technische Universitat Dresden, Germany, CBI Publication No:304, pp. 321–332.

Aziz, Z., Anumba, C.J., Ruikar, D., Carrillo, P.M., and Bouchlaghem, D.N. (2006). “Intelligent Wireless Web Services for Construction – A review of Enabling Technologies,” Automation in Construction, Vol. 15(2), Elsevier, 113–123.

Behzadan A. H., Aziz Z., Anumba C., and Kamat V. R. (2008). “Ubiquitous Location Tracking for Context Specific Information Delivery on Construction Sites,” Automation in Construction, Vol. 17(6), Elsevier Science, New York, NY, pp. 737–748.

Fathi, M.S., Anumba, C.J., and Carrillo, P. (2006). “Context Awareness in Construction Management – Key Issues and Enabling Technologies,” Proc. of the Joint Int. Conference on Construction Culture, Innovation, and Management (CCIM’06), Dubai, UAE, pp. 425–435.

Hart, S.G., and Staveland, L.E. (1988). “Development of NASA TLX (Task Load Index): Results of Empirical and Theoretical Research,” Human Mental Workload, Edited by Hancock, P.A., and Meshkati, N., Elsevier, Amsterdam, The Netherlands, pp. 139–183.

Khoury, H. M. (2009). "Context Aware Information Access and Retrieval for Rapid On-Site Decision Making in Construction, Inspection and Maintenance of Constructed Facilities,” Ph.D. Dissertation, Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI.

Khoury, H. M., and Kamat, V. R. (2009). "High-Precision Identification of Contextual Information in Location-Aware Engineering Applications,” Advanced Engineering Informatics, Vol. 23 (4), Elsevier Science, New York, NY, pp. 483–496.

Kortuem, G., Bauer, M., and Segall, Z. (1999). “NETMAN: the Design of a Collaborative Wearable Computer System,” Mobile Networks and Applications, Vol. 4(1), pp. 49–58.

Leckie, C. and Dale, M. (1997). “Locating Faults in Tree-Structured Networks,” Proc. of the 15th Int. Joint Conference on Artificial Intelligence (IJCAI), Nagoya, Japan, pp. 434–439.

Omar, B., and Ballal, T. (2009). “Intelligent Wireless Web Services: Context-aware Computing in Construction-logistics Supply Chain,” Journal of Information Technology in Construction (ITcon), Vol. 14, pp. 289–308.

Pascoe, J., Morse, D.R., and Ryan, N.S. (1998). “Developing Personal Technology for the Field,” Journal of Personal and Ubiquitous Computing, Vol. 2(1), Springer, London, pp. 28–36.

Saidi, K., Cheok, G., Franaszek, M., Brown, C., Swerdlow, J., Lipman, R., Katz, I., Golparvar-Fard, M., Goodrum, P., Akula, M., Dadi, G., and Ghadimi, B. (2011). “Report on the Development and Use of the NIST Intelligent and Automated Construction Job Site Testbed,” National Institute of Standards and Technology (NIST), Gaithersburg, MD.

Skibniewski, M.J., and Jang, W.S. (2006). “Ubiquitous Computing: Object Tracking and Monitoring in Construction Processes Utilizing ZigBee Networks,” Proc. of the 23rd Int. Symp. on Automation and Robotics in Construction (ISARC), Tokyo, Japan, pp. 287–292.

Son, H., Kim, Ch., Kim, H., Choi, K.-N., and Jee, J.-M. (2008). “Real-time Object Recognition and Modeling for Heavy-Equipment Operation,” Proc. 25th Int. Symp. on Automation and Robotics in Construction, Vilnius, pp. 232–237.

Teizer, J.D., Venugopal, M., and Walia, A. (2008). “Ultra Wideband for Automated Real-time Three-Dimensional Location Sensing for Workforce, Equipment and Material Positioning and Tracking,” Proc. of the 87th Transportation Research Board Annual Meeting, Washington, D.C.

Zou, W., Ye, X., and Chen, Z. (2006). “A brief Review on Application of Mobile Computing in Construction,” 1st Int. Multi-Symp. on Computer and Computational Sciences (IMSCCS’06), Vol. 2, Zhejiang University, Hangzhou, China, pp. 657–661.

Appendices

Appendix A

Implementation of the Integrated Tracking System

A.1 Integration of Real-Time Kinematic-Global Positioning System and Personal Dead Reckoning

To implement an Integrated Tracking System (ITS) based on Real-Time Kinematic Global Positioning System (RTK-GPS) and Personal Dead Reckoning (PDR), the mobile user is suited with the RTK-GPS and PDR hardware, as shown in Figure A.1. The RTK-GPS receiver is fixed on the user's hard hat and the GPS measurement unit is placed in the body suit. The PDR system's (Ojeda and Borenstein 2007) Inertial Measurement Unit (IMU) is strapped to the user's foot and the PDR computing unit is clipped to a pocket. The tablet-PC carried by the user is installed with the ITS software (written in the C++ programming language) and acts as the computation center. The difference in the location of the RTK-GPS and PDR sensing units is resolved by applying the user's height—where deemed necessary—in the ITS algorithm.

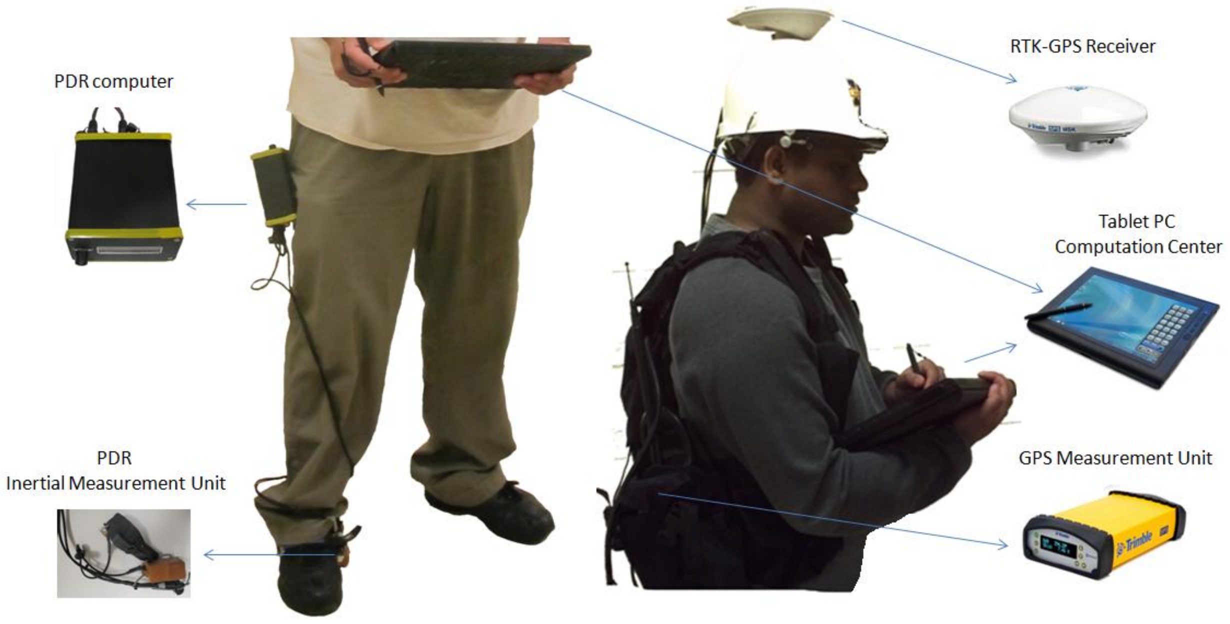
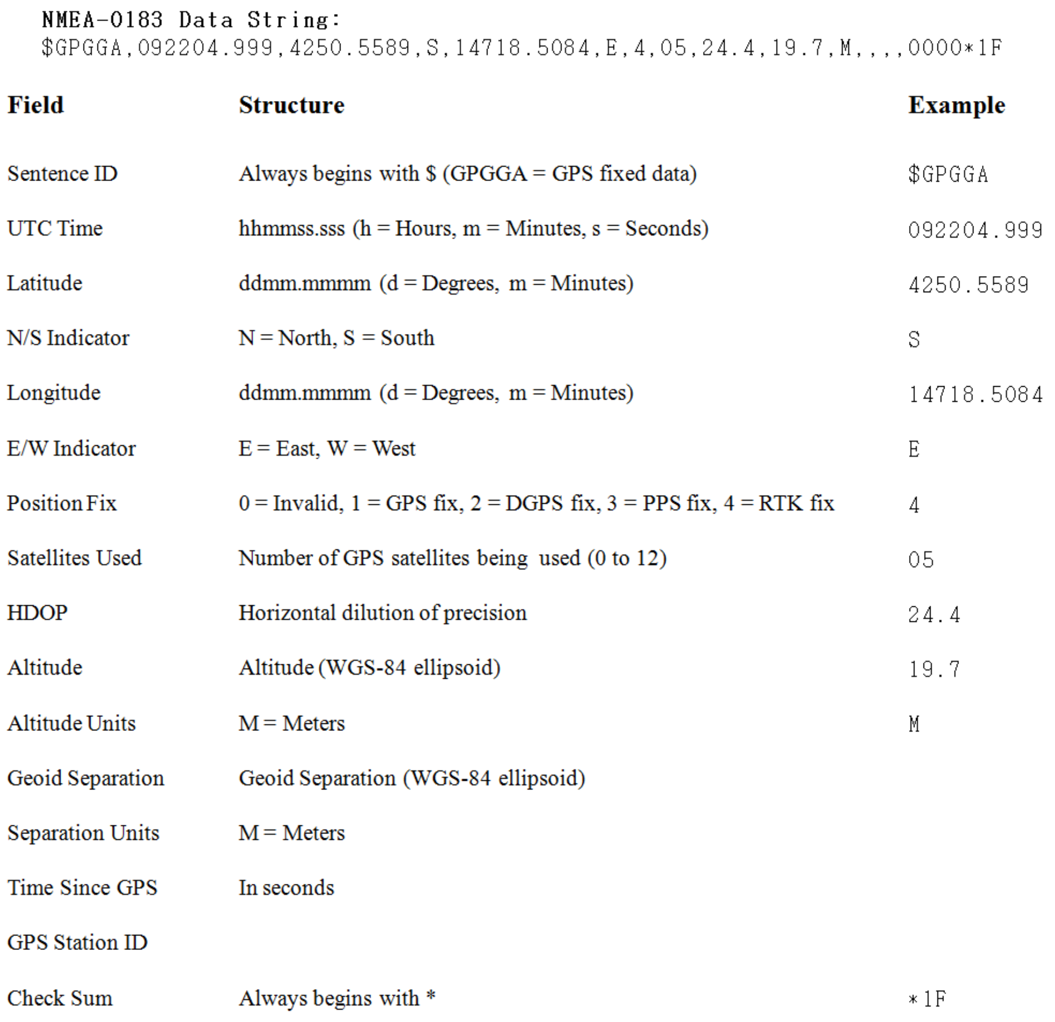


Figure A.1: RTK-GPS and PDR system hardware fitted on a mobile user

The RTK-GPS unit communicates—either through a wired connection or wirelessly—with the computation center through a serial port. The serial port communication settings are 4800 kbps, 8 data bits, no parity, and 1 or 2 stop bits. The RTK-GPS unit communicates GPS data using the NMEA-0183 format—an interface specification for GPS that was developed by the National Marine Electronics Association (NMEA). The ITS software expects to receive GPS data as packets encoded in the NMEA-0183 format. Each packet with GPS fixed data is comprised of 16 fields (of fixed size) separated from one another by a single character, as shown in Figure A.2.



NMEA-0183 Data String:

```
$GPGGA,092204.999,4250.5589,S,14718.5084,E,4,05,24.4,19.7,M,,,,0000*1F
```

Field	**Structure**	**Example**
Sentence ID	Always begins with $ (GPGGA = GPS fixed data)	$GPGGA
UTC Time	hhmmss.sss (h = Hours, m = Minutes, s = Seconds)	092204.999
Latitude	ddmm.mmmm (d = Degrees, m = Minutes)	4250.5589
N/S Indicator	N = North, S = South	S
Longitude	ddmm.mmmm (d = Degrees, m = Minutes)	14718.5084
E/W Indicator	E = East, W = West	E
Position Fix	0 = Invalid, 1 = GPS fix, 2 = DGPS fix, 3 = PPS fix, 4 = RTK fix	4
Satellites Used	Number of GPS satellites being used (0 to 12)	05
HDOP	Horizontal dilution of precision	24.4
Altitude	Altitude (WGS-84 ellipsoid)	19.7
Altitude Units	M = Meters	M
Geoid Separation	Geoid Separation (WGS-84 ellipsoid)	
Separation Units	M = Meters	
Time Since GPS	In seconds	
GPS Station ID		
Check Sum	Always begins with *	*1F

Figure A.2: NMEA-0183 format for GPS communication

When the GPS fixed data packet arrives at the serial port, the ITS parses the received information. The quality of the data is considered acceptable if the position fix is RTK and if at least 4 satellites were used to determine GPS coordinates. If the data quality is considered acceptable, the ITS parses the packet and extracts the RTK-GPS coordinates, as illustrated in the code snippet shown below.

```
System::Void RTK::OnEventCharReceivedFunc(System::Object ^sender,
                                SciCom::PortController::EventCharReceivedEventArgs ^e)
{
       array<System::String^> ^SplitArray = gcnew array<System::String^>(10);

       // If the sentence contains GPS fixed data, parse it out
       if(e->strReadBuffer->StartsWith("$GPGGA"))
       {
              System::String ^split = ",";
              array<wchar_t> ^ delimiter = split->ToCharArray();
              SplitArray = e->strReadBuffer->Split(delimiter);

              // Extract fix quality and number of satellites locked
              _fixQuality = SplitArray[6];
              _numSatellite = SplitArray[7];

              // Check whether the data quality is acceptable
              if (System::Convert::ToInt16(_fixQuality) < 2 ||
                                     System::Convert::ToInt16(_numSatellite )< 4)
              {
                     // GPS coordinates are of unacceptable quality
                     _strLat = " ";
                     _strLon = " ";
                     _strAlt = " ";
              }
              else
              {
                     // Extract GPS coordinates
                     _strLat = System::String::Concat(SplitArray[2], " ",
                                                                  SplitArray[3]);
                     _strLon = System::String::Concat(SplitArray[4], " ",
                                                                  SplitArray[5]);
                     _strAlt = System::String::Concat(SplitArray[9], " ",
                                                                  SplitArray[10]);
                     _strTime = SplitArray[1];
              }

       }
}
```

The PDR system transmits location updates at every footfall (i.e., about once per second). The term “footfall” signifies the period of time during which the IMU-equipped foot, in particular the ball of the foot, is in contact with the ground. In the event that the mobile user remains stationary, the PDR system continues to send location updates once per second (Ojeda and Borenstein 2007). Data is simultaneously transmitted through a wired connection and an internal serial radio. The serial port communication settings are 115,200 kbps, 8 data bits, no parity, and 1 stop bit. The data encodes the user’s location coordinates in the user’s starting body coordinate system, which is shown in Figure A.3. The y-axis of the PDR coordinate system is in the direction of the user’s motion during the initial calibration phase of the walk (Ojeda and Borenstein 2007).

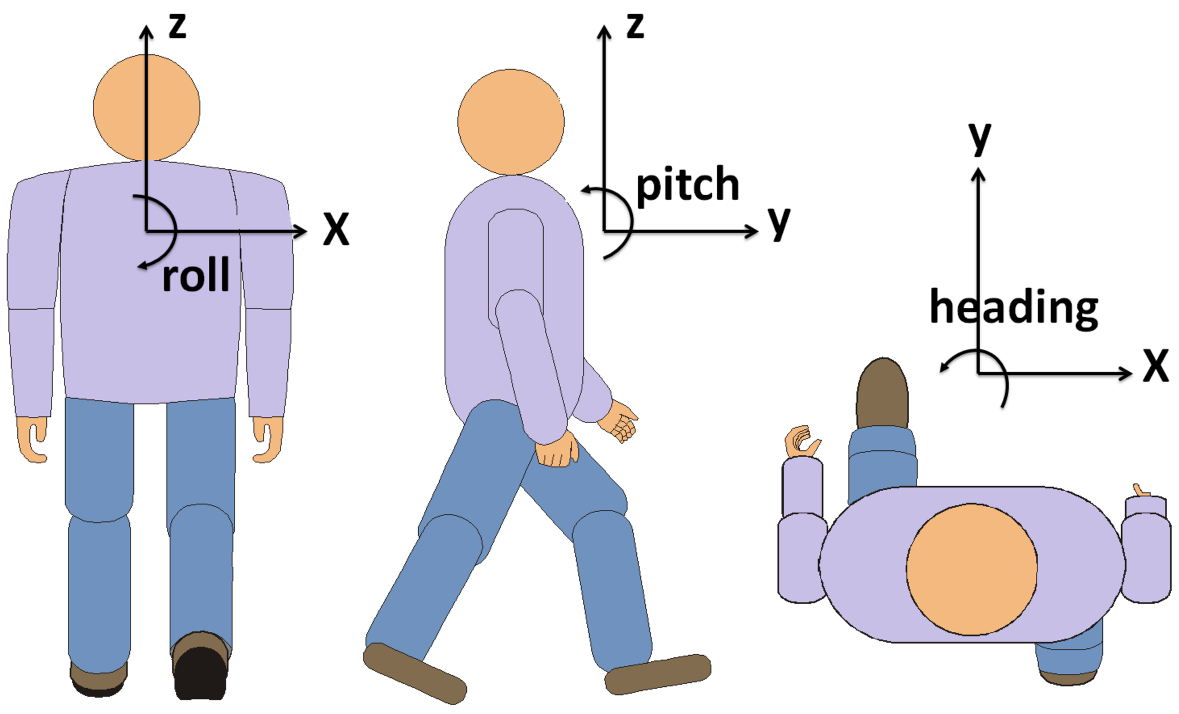
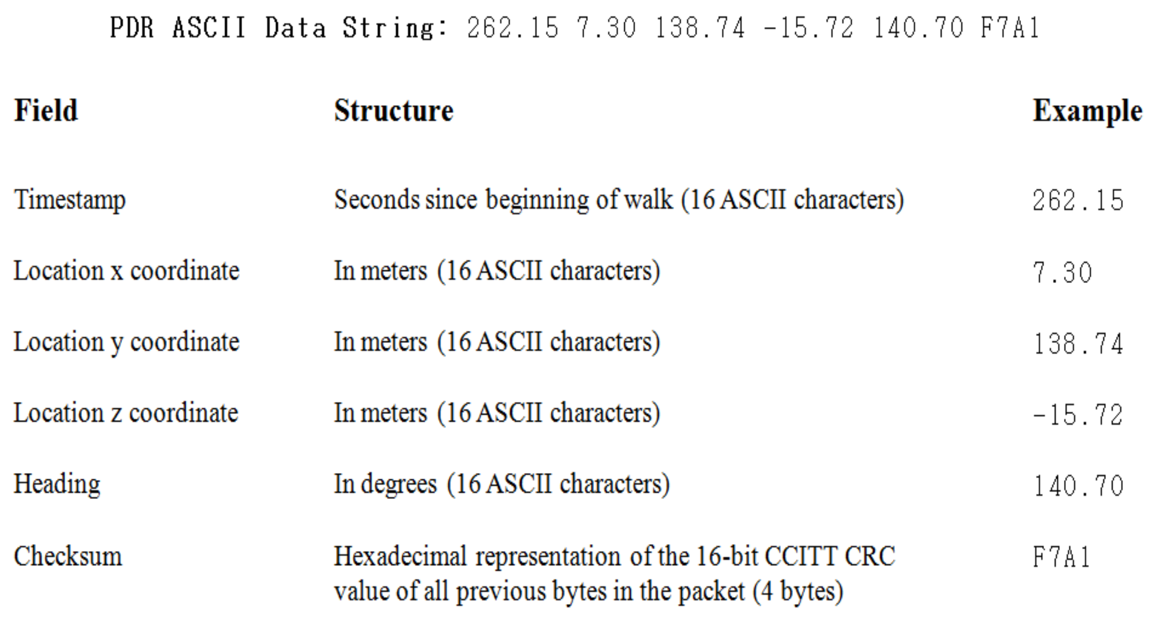


Figure A.3: User’s starting body coordinate system used to define PDR coordinate system

The PDR communicates the location coordinates using the American Standard Code for Information Interchange (ASCII) scheme. Each packet is comprised of 6 fields (of fixed size), and the fields are separated from one another by a single space character. Additional space characters are used to pad data to fill the entire field. A new line character and a carriage return character after the checksum identify the end of the packet (Ojeda and Borenstein 2007).



PDR ASCII Data String: `262.15 7.30 138.74 -15.72 140.70 F7A1`

Field	Structure	Example
Timestamp	Seconds since beginning of walk (16 ASCII characters)	262.15
Location x coordinate	In meters (16 ASCII characters)	7.30
Location y coordinate	In meters (16 ASCII characters)	138.74
Location z coordinate	In meters (16 ASCII characters)	-15.72
Heading	In degrees (16 ASCII characters)	140.70
Checksum	Hexadecimal representation of the 16-bit CCITT CRC value of all previous bytes in the packet (4 bytes)	F7A1

Figure A.4: ASCII-encoded PDR data format

When the PDR data packet arrives at the serial port, the ITS parses the information and extracts the timestamp, location, and heading data information, as illustrated in the code snippet shown below.

```
// Upon receiving a PDR data packet

System::Void PDR_Reader::OnEventCharReceivedFunc(System::Object ^sender,
                            SciCom::PortController::EventCharReceivedEventArgs ^e)
{
       // Parse PDR data string by space (“ ”) delimiter

       array<System::String^> ^SplitArray = gcnew array<System::String^>(5);

       System::String ^split = " ";

       array<wchar_t> ^ delimiter = split->ToCharArray();

       SplitArray = e->strReadBuffer->Split(delimiter);

       string _str;

       this->DotNetStringToStdString(e->strReadBuffer, _str);

       // Convert string type to floating point type

       float fx,fy,fz,ft,fh;

       sscanf(_str.c_str(), "%*x %*d %*d %f %*f %f %f %f %f %*x",
                                                 &ft, &fx, &fy, &fz, &fh);

              _x = fx.ToString(); // x coordinate of the location
              _y = fy.ToString(); // y coordinate of the location
              _z = fz.ToString(); // z coordinate of the location
              _t = ft.ToString(); // time (in seconds) since beginning of the walk
              _h = fh.ToString(); // mobile user’s heading

}
```

The ITS then converts the PDR coordinates into GPS coordinates using Vincenty’s direct algorithm (Vincenty 1975). The implementation of Vincenty’s direct algorithm is discussed in detail in Appendix A.2. The ITS uses the user’s GPS coordinates and PDR location coordinates—after being converted to the GPS coordinate system—as inputs to the integration algorithm. The integration algorithm is discussed in detail in Section 3.3.1, and its implementation is presented in the code snippet shown below. The resulting ITS location coordinates are provided as a service to client applications such as the Widely Integrated Simulation Environment (WISE) presented in Section 3.4.1.

```
void Mujo::Integrator()
{
       // Latitude:  GPS_coordinates[ 0 ], PDR_coordinates[ 0 ], IntegratedTemp[ 0 ]
       // Longitude: GPS_coordinates[ 1 ], PDR_coordinates[ 1 ], IntegratedTemp[ 1 ]
       // Altitude:  GPS_coordinates[ 2 ], PDR_coordinates[ 2 ], IntegratedTemp[ 2 ]
       // Time (H):  GPS_coordinates[ 3 ], PDR_coordinates[ 3 ], IntegratedTemp[ 3 ]
       // Time (M):  GPS_coordinates[ 4 ], PDR_coordinates[ 4 ], IntegratedTemp[ 4 ]
       // Time (S):  GPS_coordinates[ 5 ], PDR_coordinates[ 5 ], IntegratedTemp[ 5 ]
       // ITS Source: IntegratedTemp[ 6 ] { 0 = PDR, 1 = RTK-GPS)

       int ETD; // ETD = Elementary Time Difference

       ETD = 3600*PDR_coordinates[ 3 ] + 60*PDR_coordinates[ 4 ] + PDR_coordinates[ 5
       ] - 3600*GPS_coordinates[ 3 ] - 60*GPS_coordinates[ 4 ] - GPS_coordinates[ 5 ];

       if (ETD <= 0) // GPS location is the latest location
       {
              IntegratedTemp[0] = GPS_coordinates[ 0 ];
              IntegratedTemp[1] = GPS_coordinates[ 1 ];
              IntegratedTemp[2] = GPS_coordinates[ 2 ];
              IntegratedTemp[3] = GPS_coordinates[ 3 ];
              IntegratedTemp[4] = GPS_coordinates[ 4 ];
              IntegratedTemp[5] = GPS_coordinates[ 5 ];
              IntegratedTemp[6] = 1;
       }
       else // PDR location is the latest location
       {
              // Update correction if GPS is available

              if ( 3600*Correction[ 3 ] + 60*Correction[ 4 ] + Correction[ 5 ]
                     - 3600*GPS_coordinates[ 3 ] - 60*GPS_coordinates[ 4 ] –
                                                        GPS_coordinates[ 5 ] < 0)
              {
                     Correction[ 0 ] = GPS_coordinates[ 0 ] - PDR_coordinates[ 0 ];
                     Correction[ 1 ] = GPS_coordinates[ 1 ] - PDR_coordinates[ 1 ];
                     Correction[ 2 ] = GPS_coordinates[ 2 ] - PDR_coordinates[ 2 ];
                     Correction[ 3 ] = PDR_coordinates[ 3 ];
                     Correction[ 4 ] = PDR_coordinates[ 4 ];
                     Correction[ 5 ] = PDR_coordinates[ 5 ];
              }

              // Update ITS location coordinates
              IntegratedTemp[0] = PDR_coordinates[ 0 ] + Correction[ 0 ];
              IntegratedTemp[1] = PDR_coordinates[ 1 ] + Correction[ 1 ];
              IntegratedTemp[2] = PDR_coordinates[ 2 ] + Correction[ 2 ];
              IntegratedTemp[3] = PDR_coordinates[ 3 ];
              IntegratedTemp[4] = PDR_coordinates[ 4 ];
              IntegratedTemp[5] = PDR_coordinates[ 5 ];
              IntegratedTemp[6] = 0;
       }
}
```

A.2 Vincenty's Direct Algorithm

Vincenty's direct and inverse formulae (Vincenty 1975) are two related iterative methods used in geodesy to solve the direct and inverse geodetic problems. The methods establish relationships between the geographic coordinates of two points and the distance between them, and were developed by assuming the Earth to be an oblate spheroid.

Vincenty's direct formula: This method computes the geographic location of a point that is at a given distance and azimuth from another point.

Vincenty's inverse formula: This method computes the geographical distance and azimuth between two points with known geographic locations.

A detailed discussion of Vincenty's solutions and their proofs (Vincenty 1975) is beyond the scope of this Appendix. For relatively small regions (less than 10 sq. km), the ITS assumes that the distance travelled by a user—as measured by the PDR in the x-y plane of the PDR coordinate frame—is the same as the ellipsoidal distance. The ITS converts the PDR location coordinates into GPS coordinate system by implementing Vincenty's direct algorithm, as presented in the code snippet shown below. The implementation approximates the Earth's oblate ellipsoid based on the World Geodetic System (WSG) latest standard (WSG-84) dating from 1984 and last revised in 2004.

```
void Mujo::DirectVincenty(double Lat1, double Long1, double Alt1, double X, double Y,
                                                                            double Z)
{
    // Ellipsoid definition

    const double a = 6378137; // semi-major axis of the WSG-84 ellipsoid
    const double b = 6356752.3142; // semi-minor axis of the WSG-84 ellipsoid
    const double f = 1/298.257223563; // flattening of the WSG-84 ellipsoid
    const double PI = 3.1415926535; // Yummy mathematical constant

    // Variables that need to be determined
    // Point 2's geographic coordinates

    double Lat2, sinLat2, cosLat2, tanLat2; // Latitude of Point 2

    double Long2; // Longitude of Point 2

    double Alt2; // Altitude of Point 2. Alt1 comes from input


    // Point 1's geodetic coordinates, spherical distances, and related variables

    double sinLat1, cosLat1, tanLat1; // Latitude of Point 1 (Lat1 from input)

    double U1, sinU1, cosU1, tanU1; // Reduced Lat1 (Latitude on auxiliary sphere)

    double Alpha1, sinAlpha1, cosAlpha1, tanAlpha1; // Forward azimuth at Point 1

    double Alpha, sinAlpha, cosAlpha, tanAlpha; // Azimuth at Equator

    double s; // Ellipsoidal distance between Point 1 & 2

    double z; // Azimuth of the geodesic at the equator

    double Sigma, sinSigma, cosSigma; // Arc length between Point 1 & 2 on the
                                      // ... auxiliary sphere

    double us; // us = u^2 = ((cosAlpha)^2)*(a^2 – b^2)/b^2

    double Lambda, sinLambda, cosLambda, tanLambda; // Difference in longitude on the
                                                    // ... auxiliary sphere
    double Sigma1, sinSigma1, cosSigma1, tanSigma1; // Angular distance on the sphere
                                                    // ... from the equator to Point 1
    double Sigmam, sinSigmam, cosSigmam, tanSigmam; // Angular distance on the sphere
                                                    // ... from the equator to the
                                                    // ... midpoint of the line
    double sin2Sigmam, cos2Sigmam, tan2Sigmam; // sin, cos, and tan for 2*Sigmam

    // Vincenty's internal variables

    double DeltaSigma;
    double A;
    double B;
    double C;
    double L;
```

```
     // Preliminary Equations

      Alpha1 = atan(abs(X)/abs(Y));

      // Throwing heading into trhe 4 quadrants

      if (X > 0 && Y > 0)
             Alpha1 = Alpha1 // Alpha 1 will remain the same
      else if (X < 0 && Y > 0)
             Alpha1 = 2*PI - Alpha1;
      else if (X < 0 && Y < 0)
             Alpha1 = Alpha1 + PI;
      else
             Alpha1 = PI - Alpha1;

      // More preliminary Equations

      s = sqrt(X*X + Y*Y);
      z = Z;

      // Degrees to Radian conversion

      Lat1 = Lat1*PI/180;
      Long1 = Long1*PI/180;

      // Lat1 comes from function definition
      sinLat1 = sin(Lat1);
      cosLat1 = cos(Lat1);
      tanLat1 = tan(Lat1);

      tanU1 = (1-f)*tanLat1;
      U1 = atan(tanU1);
      sinU1 = sin(U1);
      cosU1 = cos(U1);

      // Alpha1 comes from function definition
      sinAlpha1 = sin(Alpha1);
      cosAlpha1 = cos(Alpha1);
      tanAlpha1 = tan(Alpha1);

      // Forward Vincenty Equations
      // Eq. V.1 through Eq. V.10 are numbered based on Vincenty's original article
      // published in Survey Review 1975

      tanSigma1 = tanU1/cosAlpha1;             // (V.1)

      Sigma1 = atan(tanSigma1);
      cosSigma1 = cos(Sigma1);
      sinSigma1 = sin(Sigma1);

      sinAlpha = cosU1*sinAlpha1;              // (V.2)

      Alpha = asin(sinAlpha);
      cosAlpha = cos(Alpha);
      tanAlpha = tan(Alpha);
```

```
        us = cosAlpha*cosAlpha*(a*a - b*b)/(b*b);

        A = 1 + us*( 4096 + us*( -768 + us*( 320 - 175*us ) ) )/16384;          // (V.3)

        B = us*( 256 + us*( -128 + us*( 74-47*us ) ) )/1024;                    // (V.4)

        Sigma = s/(b*A);     sinSigma = sin(Sigma);      cosSigma = cos(Sigma);

        while (abs(DeltaSigma)>0.00001) // Iterate till Sigma has desired accuracy
        {
               Sigmam = (2*Sigma1 + Sigma)/2;                                   // (V.5)

               cosSigmam = cos(Sigmam);
               tanSigmam = tan(Sigmam);
               sinSigmam = sin(Sigmam);
               cos2Sigmam = cos(2*Sigmam);
               tan2Sigmam = tan(2*Sigmam);
               sin2Sigmam = sin(2*Sigmam);

               DeltaSigma = B*sinSigma*( cos2Sigmam + B*( cosSigma*( -1 +
                      2*cos2Sigmam*cos2Sigmam ) - B*cos2Sigmam*( -3 +
                      4*sinSigma*sinSigma )*(-3 + 4*cos2Sigmam*cos2Sigmam)/6 )/4 );

               Sigma = DeltaSigma + s/(b*A);                                    // (V.7)
        }

        tanLat2 = (sinU1*cosSigma + cosU1*sinSigma*cosAlpha1)/((1-
               f)*sqrt(sinAlpha*sinAlpha + (sinU1*sinSigma -
               cosU1*cosSigma*cosAlpha1)*(sinU1*sinSigma
               - cosU1*cosSigma*cosAlpha1)));                                   // (V.8)

        Lat2 = atan(tanLat2);
        sinLat2 = sin(Lat2);
        cosLat2 = cos(Lat2);

        tanLambda = (sinSigma*sinAlpha1)/( cosU1*cosSigma
                                           - sinU1*sinSigma*cosAlpha1 );       // (V.9)
        Lambda = atan(tanLambda);
        sinLambda = sin(Lambda);
        cosLambda = cos(Lambda);

        C = f*cosAlpha*cosAlpha*( 4 + f*(4-3*cosAlpha*cosAlpha) )/16;          // (V.10)

        L = Lambda - (1-C)*f*sinAlpha*( Sigma
           + C*sinSigma*( cos2Sigmam + C*cosSigma*( -1 + 2*cos2Sigmam*cos2Sigmam ) ) );

        Long2 = Long1 + L;
        Alt2 = Alt1 + z;

        // Radian to Degrees conversion
        Lat2 = Lat2*180/PI; Long2 = Long2*180/PI;

        // Insert Geographic coordinates of Point 2 in the Vincenty array
        VincentyTemp[0] = Lat2;
        VincentyTemp[1] = Long2;
        VincentyTemp[2] = Alt2;
}
```

A.3 Human Intelligence-Based Tracking System

To implement the hybrid tracking system based on integrating the PDR system with human-intelligence, the mobile user is suited with the PDR hardware, as shown in Figure A.1. The PDR system's (Ojeda and Borenstein 2007) Inertial Measurement Unit (IMU) is strapped to the user's foot and the PDR computing unit is clipped to a pocket. The tablet-PC carried by the user is installed with the hybrid tracking system software and acts as the computation center. The hybrid tracking application presents users with a visualization of their tracked location relative to their environment. The visualization application allows users to continuously correct their locations—if deemed necessary—by reconciling the presented visualization and their actual environment. The software is developed using the C++ language, and visualization features are implemented using the OpenSceneGraph (OSG) toolkit.

OSG is an open source, high performance, cross platform 3D graphics toolkit written entirely in Standard C++ and OpenGL. OSG implements scene graph data structures and provides an object-oriented framework on top of OpenGL, thus freeing developers from implementing and optimizing low level graphics calls and methods.

An OSG scene graph is a collection of nodes in a graph structure connected together through a child-parent relationship. A node is defined as a scene graph object that can be part of or can entirely comprise a scene graph. A node is a collection of fields and methods that perform specific functions, thus encapsulating the semantics of what is to be drawn. Methods applied to a parent node will affect all its child nodes. The nodes are thus ideally arranged in a hierarchical scheme that corresponds spatially and semantically to the modeled scene. A detailed discussion of the core classes underlying OSG scene graph trees (OSG 2013) is beyond the scope of this abstract. The most important nodes or core classes underlying typical OSG scene graphs are as follows (OSG 2013):

- *osg::Node:* is the base class defining all of the internal nodes in the scene graph. It provides an interface for most of the common node operations, and it mainly consists of *osg::Group* and *osg::Geode* classes.
- *osg::Group:* is a class that maintains a list of children and is used to group objects together.
- *osg::Geode:* is a "geometry node" (i.e., a leaf node on the scene graph that can hold renderable geometric data). Renderable geometric data is represented by objects from the osg::Drawable class. Thus, a Geode is a Node whose purpose is grouping Drawables.
- *osg::Drawable:* is a pure virtual base class for implementing drawable geometry. The *osg::Drawable* class contains no drawing primitives, which are instead provided by subclasses such as *osg::Geometry*.
- *osg::Transform:* is a group node for which all children are transformed by a 4x4 matrix. It is often used for positioning objects within a scene, for producing trackball functionality, or for animation.
- *osg::PositionAttitudeTransform:* is a Transform that sets coordinate transformation defined by a 3D position vector and a rotation quaternion.
- *osg::StateSet:* is a core class that stores a set of models and attributes that encapsulate a set of OpenGL states. Each Drawable and each Node has a reference to a StateSet. These StateSets can be shared between different Drawables and Nodes.

- *osg::Texture:* is a pure virtual base class that encapsulates OpenGl texture functionality common to the various types of OSG textures.

The visualization application is developed as an OSG scene graph with Nodes to represent geometric models of the user and environment, as shown in Figure A.5. The mobile user is modeled as a PositionAttitudeTransform node whose 3D position vector is determined by the PDR system, thus allowing for animation effects.

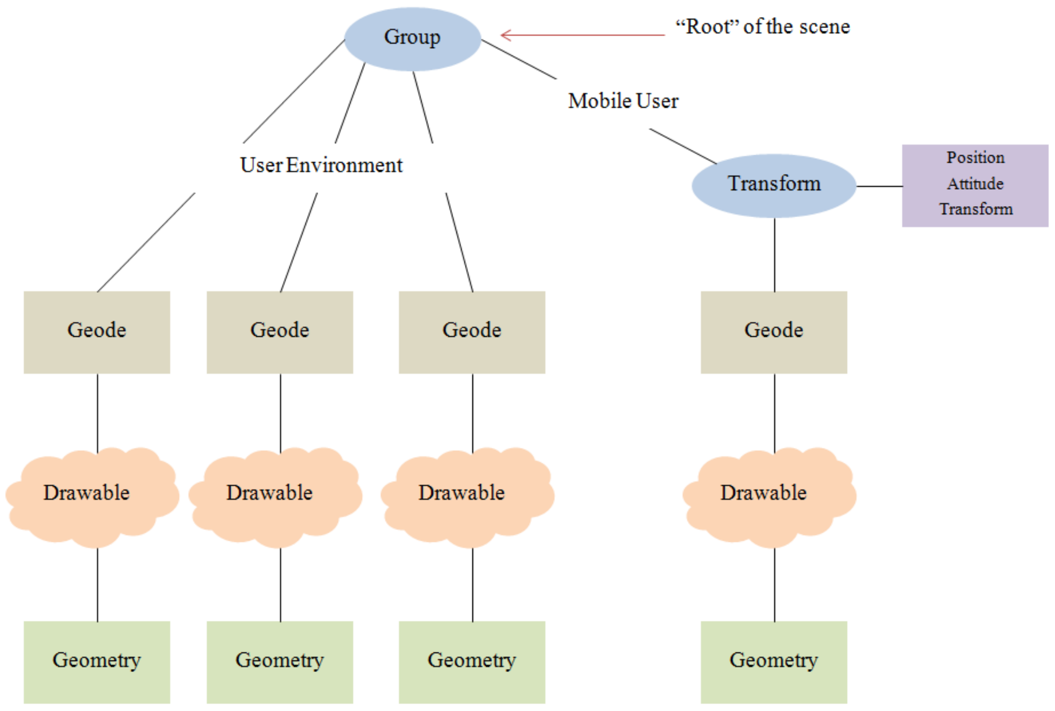


Figure A.5: Scene graph of the hybrid tracking system's visualization application

The hybrid tracking system algorithm presents the user with a continuous visualization of the aforementioned scene graph using the OSG classes and methods, as shown in the code snippet presented below. Upon receiving a new PDR packet at the serial port, the algorithm parses the

PDR data, interprets the location of the user, and applies appropriate transformation to the nodes pertaining to the mobile user, which is also illustrated in the code snippet shown below.

```
        // Create rootnode.
        osg::Group* rootnode = new osg::Group();

        // Create the environment nodes and attach them to the rootnode.
        osg::Node* modelEnvironment = createEnvironment();
        rootnode->addChild(modelEnvironment);

        // Initializing the viewer.
        osgViewer::CompositeViewer compositeViewer;
        osgViewer::View* primeView = new osgViewer::View;

        // Define primeView
        primeView->setName("Chase Cam");
        primeView->setSceneData(rootnode);
        primeView->setUpViewOnSingleScreen(0);
        primeView->getCamera()->setViewport(new osg::Viewport(50,50,500,500));
        primeView->setUpViewInWindow(600,600,400,300,0);
        compositeViewer.addView(primeView);

        // Setting up event handlers
        KeyboardEventHandler* correctionEventHandler = new KeyboardEventHandler();
        primeView->addEventHandler(correctionEventHandler);

        // Initialize user location coordinates

        double userX = 0.0;
        double userY = 0.0;
        double userZ = 0.0;

        // Initialize location corrections

        double positionCorrectionX = 0.0;
        double positionCorrectionY = 0.0;
        double positionCorrectionZ = 0.0;

        // Create user node as positionattitudetranform

        osg::Node* trackedUser = createTrackedUser(userX, userY, userZ);
        osg::PositionAttitudeTransform* trackedUserPAT = new
                                          osg::PositionAttitudeTransform();
        trackedUserPAT->addChild(trackedUser);
        rootnode->addChild(trackedUserPAT);

        // Set parameters to accept PDR data from serial port

        PDR::PDR_Reader^pdrReader = gcnew PDR::PDR_Reader();
        pdrReader->Initialize(PDR_ComPortID_String, 115200);
        array<double>^ pdrInfo = gcnew array<double>(5);
        while (EXIT == false)
        {
               // Set user's location based on the tracked PDR location

               trackedUserPAT->setPosition(osg::Vec3d(userX, userY, userZ));

               // Move 3rd person camera along with user
               // This particular camera tracks the user from the top view

               primeView->getCamera()->setViewMatrixAsLookAt(osg::Vec3(userX,
                      userY, userZ + 40), osg::Vec3(userX, userY, userZ),
                      osg::Vec3(0.0, 1.0, 0.0));

               // Run optimization over the scene graph
               osgUtil::Optimizer optimzer;
               optimzer.optimize(rootnode);

               // Realize compositeviewer and view the frame
               compositeViewer.realize();
               compositeViewer.frame();

               // Correct the user's location if user updates them via keyboard event
               // handlers

               if (UPKEY == true)
               {
                      positionCorrectionY = positionCorrectionY + 0.2;
                      UPKEY = false;
               }
               if (DOWNKEY == true)
               {
                      positionCorrectionY = positionCorrectionY - 0.2;
                      DOWNKEY = false;
               }
               if (LEFTKEY == true)
               {
                      positionCorrectionX = positionCorrectionX - 0.2;
                      LEFTKEY = false;
               }
               if (RIGHTKEY == true)
               {
                      positionCorrectionX = positionCorrectionX + 0.2;
                      RIGHTKEY = false;
               }

               // Update userX, userY, userZ here from latest PDR data packet

               pdrInfo = pdrReader->getPDRInfo();
               userX = pdrInfo[0];
               userY = pdrInfo[1];
               userZ = pdrInfo[2];

               // Apply corrections to the user's location

               userX = userX + positionCorrectionX;
               userY = userY + positionCorrectionY;
               userZ = userZ + positionCorrectionZ;
        }
```

The user communicates the location corrections using a keyboard and pressing appropriate keys to move the user's model/avatar relative to the environment. The algorithm communicates the user's corrective actions through the implementation of a keyboard event handler function, as illustrated in the code snippet shown below.

```
// Event handler functions
bool myKeyboardEventHandler::handle(const osgGA::GUIEventAdapter& ea,
                                                        osgGA::GUIActionAdapter& aa)
    {
       switch(ea.getEventType())
       {
       case(osgGA::GUIEventAdapter::KEYDOWN):
          {
             switch(ea.getKey()) // WSAD "gaming keys" are used to move the avatar
             {
                    case 'w':
                           UPKEY = true;
                           return false;
                           break;

                    case 's':
                           DOWNKEY = true;
                           return false;
                           break;

                    case 'a':
                           LEFTKEY = true;
                           return false;
                           break;

                    case 'd':
                           RIGHTKEY = true;
                           return false;
                           break;

                    default:
                           return false;
             }
          }

       default:
          return false;
       }
    }
```

A.4 References

Ojeda, L., and Borenstein, J. (2007). "Personal Dead Reckoning System for GPS-denied Environments," IEEE International Workshop on Safety, Security and Rescue Robotics, Institute of Electrical and Electronics Engineers (IEEE), Rome, Italy.

OSG (2013). "OpenSceneGraph" – Website of the OpenSceneGraph (OSG) Project <http://www.openscenegraph.org/>, as accessed on 28th April, 2013.

Appendix B

FLoc: An Algorithm Adopted for Locating Faults in HVAC Distribution Networks

B.1 Minimum Message Length Principle

The Minimum Message Length (MML) principle is an information theory method used to evaluate the effectiveness of competing theories (T_i) in explaining a given set of measurements (M) observed in the world (Georgeff and Wallace 1984; Oliver and Hand 1994; Wallace and Boulton 1968). The efficiency of a given theory in explaining a set of measurements is determined by constructing a message that describes both the theory and the measurements in terms of the theory. The MML principle considers one theory (T_1) to be better than another (T_2) if the total length of the message using T_1 is shorter than the message using T_2.

The MML principle argues that a theory helps explain measurements, and thus the description of the measurements becomes shorter than when no theory is used. The addition of the theory to the message description accounts for the complexity of the theory. Thus, if a theory is too simplistic, the theory's description will be short, however, the measurement's description will be long. Increased complexity in the theory introduces structure to the measurements, thus increasing the length of the theory's description while shrinking the measurement's description. However, overly complex theories result in the increased length of the theory's description overshadowing any decrease in the length of the measurement's description (Leckie and Dale 1997).

The complete message is encoded as a binary string by employing Huffman code schema. Thus, the length of the measurements becomes equal to the probability of the measurements given the theory (Leckie and Dale 1997), as shown in Eq. (B.1).

$$\text{length}(M) = -\log_2 P(M|T_i) \tag{B.1}$$

Similarly, the length of the optimum code describing the theory is equal to the probability of the theory, estimated a priori, as shown in Eq. (B.2).

$$\text{length}(T_i) = -\log_2 P(\text{Ti}) \tag{B.2}$$

Thus, the length of the message that combines the theory's and the measurement's descriptions is estimated by adding Eq. B.1 and Eq. B.2, as shown in Eq. (B.3).

$$\text{length}(M + T_i) = -\log_2[P(M|T_i)P(\text{Ti})] \tag{B.3}$$

Applying Bayes' theorem, the length of the combined message can be calculated as shown in Eq. (B.4).

$$\text{length}(M + T_i) = -\log_2[P(T_i|M)P(\text{M})] \tag{B.4}$$

As *P(M)* is constant for an observed set of measurements, the length of the combined message is minimized by the theory that has the highest probability given the measurement set (Leckie and Dale 1997). In practice, the MML principle does not require the construction of the message since the MML principle relies on the length of the constructed message rather than the actual message (Oliver and Hand 1994). The MML principle provides a method for comparing multiple theories explaining the observed measurement set, and thus addresses the two key issues of the network diagnosis problem: deciding what faults are appropriate, and estimating the effectiveness of these faults in explaining the observed measurement set.

B.2 FLoc Algorithm for Locating Faults

Based on the MML principle, the FLoc algorithm (Leckie and Dale 1997) localizes the nodes most probably at fault for a given set of observed measurements. FLoc assumes that the input HVAC network tree can have an arbitrary number of nodes N and an arbitrary number of children per node. Normally functioning nodes are considered to be not at fault (F^{OK}). The algorithm classifies faults at a node n_i into two types: the first type of fault (F^A) corresponds to major failure and affects all nodes downstream; the second type of fault (F^L) corresponds to localized failure and affects only the leaf nodes connected directly to the faulty node.

Each leaf node has a measurement m—reflecting the observed HVAC performance—that is affected by the state of the nodes upstream of the leaf. As described in Section 5.4.3, the range $[m^L, m^U]$ of the leaf node performance measurement is [0, 1]. As mentioned in Section 5.4.4, for each node, the probability density function $f^F(m)$ describes the distribution of HVAC performance measurement values m when the leaf node is affected by a fault, either directly or indirectly. The function $f^F(m)$ is determined by observing the reported performance metrics over several historic instances of HVAC failure. The FLoc algorithm uses the same distribution function $f^F(m)$ for both types of fault (F^A and F^L). For reasons mentioned in Section 5.4.4, the probability $P^N(m)$ of the possible values of HVAC performance measurement m for any leaf node when the leaf node is not affected by a fault is given by Eq. (B.5).

$$P^N(m) = \begin{cases} 1 & if\ m = 0 \\ 0 & if\ m \neq 0 \end{cases} \quad \text{(B.5)}$$

By reconciling the HVAC performance measurements observed at leaf nodes with their probability functions, the FLoc algorithm aims to assign one of the three labels (F^{OK}, F^A, F^L) to each node in the tree (Leckie and Dale 1997).

B.3 Message Structure

The complete description of an explanation for an observed measurement set includes the network structure, fault locations, and the observed measurements corresponding to these fault locations. Since a network tree structure is fixed, it can be ignored as a constant overhead that has no effect in minimizing the message length. Thus, the effective portion of the message consists of two parts: the faults hypothesized by the theory and the observed measurements (Leckie and Dale 1997).

Faults: When describing hypothesized faults, the FLoc algorithm assumes that the default status of each node is no fault (F^{OK}). Thus, the algorithm only encodes the unique identifier and the fault type for the nodes suspected to be at fault. The unique identifier for each node is $\log_2 N$ bits long, where N is the number of nodes in the tree. If the faulty node is a leaf node, then the type of fault becomes redundant. However, if the faulty node is not a leaf node, one additional bit is required to indicate the type of fault (F^A or F^L) (Leckie and Dale 1997).

Measurements: Each observed HVAC performance measurement is described by considering whether or not the leaf node is affected by a fault and using the corresponding probability function ($f^F(m)$ if the node is affected by a fault and $f^N(m)$ if the node is unaffected by faults). Each measurement *m* is encoded to a finite accuracy ε (= 0.1) by dividing the measurement range into a set of subintervals of width ε (Wallace and Boulton 1968). The measurement m_i at node n_i is approximated as m_i*, where m_i* is the midpoint of the interval containing m_i (Leckie and Dale 1997). Thus, the probability of m_i is approximated as shown in Eqs. (B.6) and (B.7).

$$P(m_i|\, no\, fault) \approx \varepsilon * f^N(\mathrm{m}_\mathrm{i}^*) \tag{B.6}$$

$$P(m_i|\, fault) \approx \varepsilon * f^F(\mathrm{m}_\mathrm{i}^*) \tag{B.7}$$

Thus, the length of the message describing the measurement m_i at a leaf node n_i is given by Eqs. (B.8) and (B.9).

$$\mathrm{length}(m_i) = -\log_2[\varepsilon * f^N(\mathrm{m}_\mathrm{i}^*)] \qquad \text{if there is no fault affecting } m_i\text{, or} \tag{B.8}$$

$$\mathrm{length}(m_i) = -\log_2[\varepsilon * f^F(\mathrm{m}_\mathrm{i}^*)] \qquad \text{if there is a fault affecting } m_i \tag{B.9}$$

B.4 Assigning Fault Status Based on MML Principle

The algorithm first assigns a sequence step to each node in the network tree based on the depth (length of the path to its root) of the node within the tree. For a given measurement set, the algorithm recursively selects the most appropriate fault status for each sub-tree starting at the leaf nodes of the final sequence step, as shown in the code snippet presented below.

```
// Node status for internal node: 0 = OK, 1 = Fault type A, 2 = Fault type L
// Node status for leaf nodes: 0 = OK, 1 = Fault type A or L

int[] nodeStatus = new int[treeSize];

    // Iterate through all sequence steps in a backward pass (from deepest to root)

    for (int currentSequenceStep = maxSequenceStep; currentSequenceStep >= 1;
         currentSequenceStep--)
    {
         // Iterate through all nodes in the tree...

         for (int i = 0; i < treeSize; i++)
         {
              // ... to determine those nodes that correspond to the
              // currentSequenceStep

              if (sequenceStep[i] == currentSequenceStep)
              {
                   // The node i belongs to the currentSequenceStep

                   // Determine whether the node i is a leaf node or an internal node
```

At each leaf node, there are two possible messages: one with a fault and one without. The algorithm selects the fault status for the leaf node based on the shorter of the two messages, as shown in the code snippet below.

```
                   if (successorIDs[i][0] == 0) // The node ni is a leaf node
                   {
                        // Determine message length if leaf is not at fault

                        double lengthMessageFok = -
                           Math.Log(pdfEpsilonOfNonFaultyLeafNode(measurements[i]), 2);

                        // Determine message length if leaf is at fault

                        double lengthMessageFault = Math.Log(treeSize, 2) -
                              Math.Log(pdfEpsilonOfFaultyLeafNode(measurements[i]), 2);

                        // Set leaf node status

                        if (lengthMessageFok <= lengthMessageFault)
                            nodeStatus[i] = 0;        // No fault at leaf
                        else
                            nodeStatus[i] = 1;        // Fault at leaf
                   }
```

The algorithm works its way up the tree, as shown in the code snippet below, by selecting a fault status for each internal node n_i, based on the message lengths of each of the subtrees rooted at the children of n_i.

```
else
{
     // The node ni is an internal node

     // Calculate message length for FOK, FA, and FL to select ...
     // ... the fault status of node ni

     double lengthMessageFok = 0;
     double lengthMessageFA = 0;
     double lengthMessageFL = 0;

     for (int j = 0; j < successorIDs[i].Length; j++)
     {
         // successorIDs[i][j] is the nodeID of the children of ni.

         int childID = 0;

         for (int y = 0; y < nodeID.Length; y++)
         {
             if (nodeID[y] == successorIDs[i][j])
             {
                 childID = y;
             }
         }
```

The message length at an interior node changes based on the fault status assigned to the node.

No fault: If there is no fault at n_i (F_i^{OK}), the message length at n_i is simply the sum of the message lengths from each child of n_i, as shown in the code snippet below (Leckie and Dale 1997).

```
    // Calculate lengthMessageFok

    if (successorIDs[childID][0] == 0)
    {
    // Child is a leaf node

        if (nodeStatus[childID] == 1 || nodeStatus[childID] == 2)
        {
            lengthMessageFok = lengthMessageFok
                  - Math.Log(pdfEpsilonOfFaultyLeafNode(measurements[childID]), 2);
        }
        else
        {
            lengthMessageFok = lengthMessageFok
               - Math.Log(pdfEpsilonOfNonFaultyLeafNode(measurements[childID]), 2);
        }
    }
    else
    {
         // Child is not a leaf node and has subtree

         // Iterate through all nodes to ...
         for (int k = 0; k < treeSize; k++)
         {
             // ... determine nodes in the subtree of successorIDs[i][j]

             if (subtreeContainmentCheck(nodeID, successorIDs,
                 successorIDs[i][j], nodeID[k]))
             {
                  // Add faults if fault exist

                  if (nodeStatus[k] == 1 || nodeStatus[k] == 2)
                  {
                       if (successorIDs[k][0] == 0) // nk is a leaf node
                           lengthMessageFok = lengthMessageFok
                                                       + Math.Log(treeSize, 2);
                       else // nk is not a leaf node
                           lengthMessageFok = lengthMessageFok
                                                       + Math.Log(treeSize, 2) + 1;
                  }

                  // Add measurements if nk is a leaf node
                  if (successorIDs[k][0] == 0)
                  {
                       // Fault exists at leaf node
                       if (nodeStatus[k] == 1 || nodeStatus[k] == 2)
                           lengthMessageFok = lengthMessageFok
                     - Math.Log(pdfEpsilonOfFaultyLeafNode(measurements[k]), 2);

                       // Fault does not exist at leaf node
                       else if (nodeStatus[k] == 0)
                           lengthMessageFok = lengthMessageFok
                     - Math.Log(pdfEpsilonOfNonFaultyLeafNode(measurements[k]), 2);
                  }
             }
         }
    }
```

Fault affecting all children: If the status assigned to node n_i is the fault status F_i^A, then this fault overrides all of the faults that were previously assigned at any node in the subtree rooted at n_i. As shown in the code snippet below, the corresponding message comprises the new fault label F_i^A at n_i and the measurements from all leaves downstream from n_i using the appropriate probability distribution ($f^F(m)$) (Leckie and Dale 1997).

```
// Calculate lengthMessageFA
// Describe the fault at node ni
lengthMessageFA = lengthMessageFA + Math.Log(treeSize, 2) + 1;

// Describe the measurements in the subtree rooted at ni

for (int j = 0; j < successorIDs[i].Length; j++)
{
     // successorIDs[i][j] is the nodeID of the children of ni.

     int childID = 0;
     for (int y = 0; y < nodeID.Length; y++)
     {
          if (nodeID[y] == successorIDs[i][j])
              childID = y;
     }

     if (successorIDs[childID][0] == 0)      // Child is a leaf node
          lengthMessageFA = lengthMessageFA
             - Math.Log(pdfEpsilonOfFaultyLeafNode(measurements[childID]), 2);
     else
     {
          // Iterate through all nodes to ...

          for (int k = 0; k < treeSize; k++)
          {
               // ... determine those nodes that are in the subtree

               if (subtreeContainmentCheck(nodeID, successorIDs,
                                            successorIDs[i][j], nodeID[k]))
               {
                    // Add measurements if nk is a leaf node

                    if (successorIDs[k][0] == 0)
                        lengthMessageFA = lengthMessageFA
                   – Math.Log(pdfEpsilonOfFaultyLeafNode(measurements[k]), 2);
               }
          }
     }
}
```

Fault affecting only leaf children: If the status assigned to node n_i is the fault status $\mathrm{F_i}^{\mathrm{L}}$, then this fault overrides all of the faults that were previously assigned at any leaf children of n_i. As shown in the code snippet below, the corresponding message comprises the new fault label F_i^L at n_i, the measurements from all of the leaf children of n_i using the appropriate distribution ($f^F(m)$), and the messages from each of the non-leaf children of n_i (Leckie and Dale 1997).

```
// Calculate lengthMessageFL for ni

// Describe the fault at node ni

lengthMessageFL = lengthMessageFL + Math.Log(treeSize, 2) + 1;

// Describe faults and measurements in subtree rooted at ni

for (int j = 0; j < successorIDs[i].Length; j++)
{
     // successorIDs[i][j] is the nodeID of the children of ni.

     int childID = 0;

     for (int y = 0; y < nodeID.Length; y++)
     {
          if (nodeID[y] == successorIDs[i][j])
              childID = y;
     }

     // Child is a leaf node

     if (successorIDs[childID][0] == 0)
     {

          if (nodeStatus[childID] == 1 || nodeStatus[childID] == 2)
              lengthMessageFL = lengthMessageFL
                 - Math.Log(pdfEpsilonOfFaultyLeafNode(measurements[childID]), 2);

          else
              lengthMessageFL = lengthMessageFL
              - Math.Log(pdfEpsilonOfNonFaultyLeafNode(measurements[childID]), 2);
     }
```

```
        else // Child is not a leaf node
        {
             // Iterate through all nodes to ...

             for (int k = 0; k < treeSize; k++)
             {
                  // ... determine those nodes that are in the subtree of
                  // successorIDs[i][j]

                  if (subtreeContainmentCheck(nodeID, successorIDs,
                                         successorIDs[i][j], nodeID[k]))
                  {
                       // Add faults if fault exist
                       if (nodeStatus[k] == 1 || nodeStatus[k] == 2)
                       {
                            if (successorIDs[k][0] == 0) // nk is a leaf node
                                lengthMessageFL = lengthMessageFL
                                                  + Math.Log(treeSize, 2);

                            else // nk is not a leaf node
                                lengthMessageFL = lengthMessageFL
                                                  + Math.Log(treeSize, 2) + 1;
                       }

                       // Add measurements if nk is a leaf node

                       if (successorIDs[k][0] == 0)
                       {
                            // If fault exists at leaf node

                            if (nodeStatus[k] == 1 || nodeStatus[k] == 2)
                                lengthMessageFL = lengthMessageFL
                           - Math.Log(pdfEpsilonOfFaultyLeafNode(measurements[k]), 2);

                            else if (nodeStatus[k] == 0)
                            // Fault does not exist at leaf node
                                lengthMessageFL = lengthMessageFL
                        - Math.Log(pdfEpsilonOfNonFaultyLeafNode(measurements[k]), 2);

                       }

                  }
             }
        }
```

Once the algorithm calculates the lengths of these three messages, it selects the fault status for n_i corresponding to the shortest message, as illustrated in the code snippet shown below.

```
//// Assign nodeStatus[i]

if (lengthMessageFok <= lengthMessageFA)
{
     if (lengthMessageFok <= lengthMessageFL)

         nodeStatus[i] = 0; // ni status = FOK

     else

         nodeStatus[i] = 2; // ni status = FL

}
else
{
     if (lengthMessageFA <= lengthMessageFL)

         nodeStatus[i] = 1; // ni status = FA

     else

         nodeStatus[i] = 2; // ni status = FL

}
```

This method optimizes the fault status for the subtree rooted at n_i, but it may be viewed as suboptimal at a higher level node when the algorithm accounts for the condition of n_i's siblings. Thus, the algorithm optimizes the fault status for all nodes in the tree by traversing the tree from the deepest sequence step to the network tree's root (Leckie and Dale 1997).

B.5 References

Georgeff, M.P., and Wallace, C.S. (1984). "A General Criterion for Inductive Inference," Proc. of the 6th European Conference on Artificial Intelligence (ECAI-84), European Coordinating Committee for Artificial Intelligence (ECCAI), Pisa, Italy, pp. 473–482.

Leckie, C. and Dale, M. (1997). "Locating Faults in Tree-Structured Networks," Proc. of the 15th Int. Joint Conference on Artificial Intelligence (IJCAI), Nagoya, Japan, pp. 434–439.

Oliver, J., and Hand, D. (1994). “Introduction to Medium Encoding Inference,” Technical Report 205, Department of Computer Science, Monash University, Clayton, Australia.

Wallace, C.S., and Boulton, D.M. (1968). “An Information Measure for Classification,” Computer Journal, British Computer Society (BCS), Vol. 11(2), pp. 185–194.

Appendix C

GPS Registration, Containment, and Prioritization Algorithms

C.1 GPS Registration Algorithm

As described in Section 6.3.5, the context-aware framework registers the mobile inspector's GPS coordinates to the designated common frame of reference. This section presents the algorithm that is used to register the mobile inspector's GPS coordinates to the common coordinate frame. This registration algorithm requires only a single transformation computation—to be input every time the user acquires new GPS coordinates—in contrast to the iterative methodology of Vincenty's inverse algorithm, thus resulting in a smaller consumption of computational and battery power. The accuracy of the proposed registration algorithm is comparable to that of Vincenty's inverse algorithm. The remainder of this section presents a discussion on the formulation of the proposed registration algorithm. The author presents the mathematics of registering the coordinates of the points that represent the inspector's eyes, the near plane center, and the far plane center—as shown in Figure 6.5—to the common frame of reference.

First, we present the following notations and definitions:

Bridge Frame of Reference (BFR): The common coordinate frame to which the CAD models and the tracking technologies are registered.

Cardinal Coordinate Frame (CCF): The localized linear frame of reference that defines the cardinal directions. The frame defines each coordinate point as (North-South, East-West, Up-Down).

Global Positioning System Frame of Reference (GPS frame): The coordinate frame in which the GPS coordinates of the mobile inspector are defined. The frame defines each coordinate point as

(Longitude, Latitude, Altitude). The GPS frame is not the same as the cardinal coordinate frame, although—in a localized region—the axis of the GPS frame aligns with the axis of the cardinal coordinate frame.

Inspector's Frame of Reference (IFR): The IFR is an orthogonal coordinate frame with its origin fixed between the inspector's eyes, with its Y axis along the inspector's line of sight, and with its Z axis aligned from the center of the inspector's eyes to the center of the forehead. The IFR is shown in Figure 6.5 (in red) and rotates along with the head orientation of the inspector.

Inspector's Location: The latitude, longitude and altitude coordinates of the inspector, obtained from the GPS device are represented $I_{GPS} = (I_{long}, I_{lat}, I_{alt})$.

The author proposes a key approximation that allows us to efficiently transform the GPS coordinates of a point into the BFR. The approximation is based on the hypothesis that within relatively small localized regions on the surface of the earth (1 square km or less), the curvature of the earth's geode is negligible for the intended application and the non-linear GPS frame can be converted to the CCF through a linear transformation with scaling factors. The scaling factors remain constant within the aforementioned small localized regions.

Consider a point 'P' in the IFR, as shown in Figure C.1, whose coordinates need to be registered in BFR. The segment 'IP' denoted as the vector $\mathbf{p} = [\mathrm{P_x} \quad \mathrm{P_y} \quad \mathrm{P_z}][I^m]$ where $[I^m]$ denotes the orthogonal basis of IFR, defines the coordinates of the point 'P' in IFR. The segment 'BI' is represented by the vector **bi**, and represents the inspector's location in BFR and the segment 'BP', represented by the vector $\mathbf{p}'$, represents the coordinates of the point of interest in BFR.

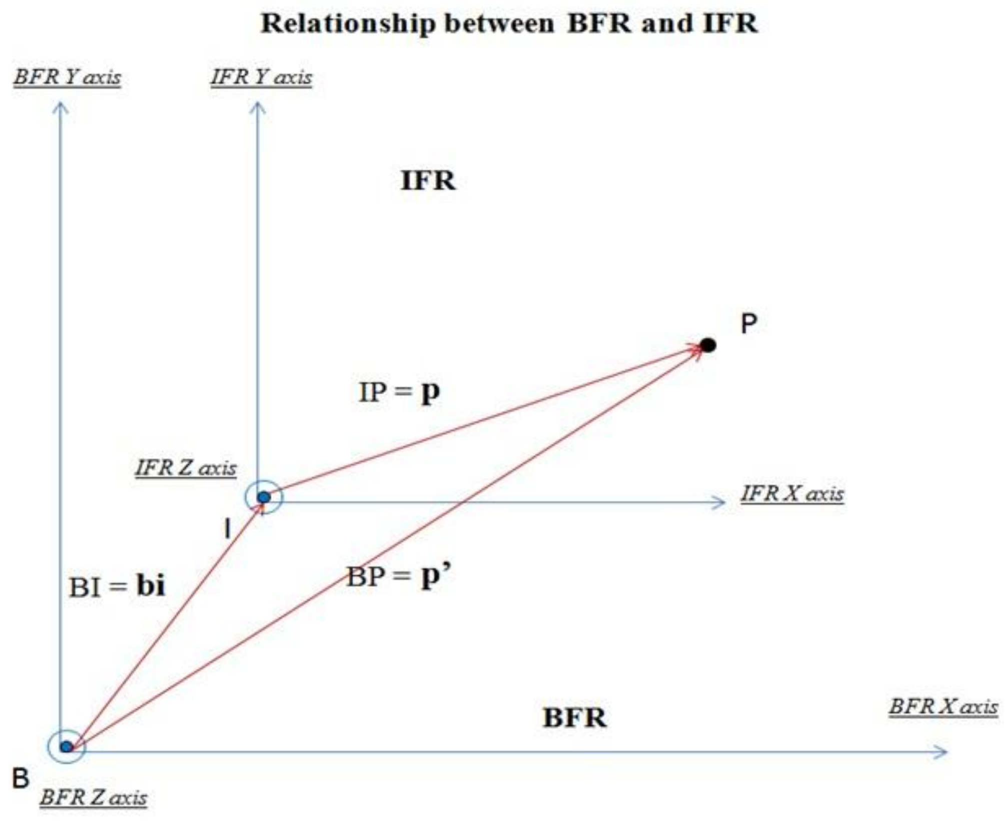


Figure C.1: Relationship between BFR and IFR to convert a point P in IFR coordinates to BFR coordinates

The vector $\mathbf{p}'$ can be written as shown in Eq. (C.1).

$$\mathbf{p}' = \mathbf{p} + \mathbf{bi} \tag{C.1}$$

The GPS coordinates of the inspector ($\mathrm{I_{long}}, \mathrm{I_{lat}}, \mathrm{I_{alt}}$) are the GPS coordinates of the origin of the IFR i.e., the point 'I'. Assuming that we know the GPS coordinates of the origin of the BFR (i.e. point 'B' ($\mathrm{B_{long}}, \mathrm{B_{lat}}, \mathrm{B_{alt}}$)), the vector **bi** can be written as shown in Eq. (C.2).

$$\mathbf{bi} = [\mathrm{I_{long}} - \mathrm{B_{long}} \quad \mathrm{I_{lat}} - \mathrm{B_{lat}} \quad \mathrm{I_{alt}} - \mathrm{B_{alt}}][G^m] \tag{C.2}$$

where $[G^m]$ is the orthogonal basis of the GPS frame of reference. Therefore, the vector $\mathbf{p}'$ can be written as shown in Eq. (C.3).

$$\mathbf{p}' = [\mathrm{P_x} \quad \mathrm{P_y} \quad \mathrm{P_z}][I^m] + [\mathrm{I_{long}} - \mathrm{B_{long}} \quad \mathrm{I_{lat}} - \mathrm{B_{lat}} \quad \mathrm{I_{alt}} - \mathrm{B_{alt}}][G^m] \tag{C.3}$$

Thus, the problem is now reduced to determining the GPS coordinates of the origin of the BFR and expressing the matrices $[I^m]$ and $[G^m]$ in terms of the matrix $[B^m]$, where $[B^m]$ is the orthogonal basis of the BFR. The matrix $[B^m]$ can be related to the matrix $[G^m]$ as shown in Eq. (C.4).

$$[B^m]\ [BG][G^m] \tag{C.4}$$

where [*BG*] is the localized linear transformation matrix that relates BFR and GPS reference frames. In order to solve for [*BG*], we provide the registration algorithm with 4 calibration points whose coordinates are known in both BFR and GPS. We represent each of these points as '$\mathrm{P_i}$', where i ∈ [1,4] and the vectors $\mathbf{p_{i\_BFR}}$ and $\mathbf{p_{i\_GPS}}$ represent the coordinates of '$\mathrm{P_i}$' in the BFR and GPS reference frames, respectively. The coordinates of '$\mathrm{P_i}$' in BFR and GPS are ($\mathrm{p_{i\_BFR\_x}}$, $\mathrm{p_{i\_BFR\_y}}$, $\mathrm{p_{i\_BFR\_z}}$) and ($\mathrm{p_{i\_GPS\_Long}}$, $\mathrm{p_{i\_GPS\_Lat}}$, $\mathrm{p_{i\_GPS\_Alt}}$), respectively. The vectors $\mathbf{p_{i\_BFR}}$ and $\mathbf{p_{i\_GPS}}$ are related as shown in Eq. (C.5).

$$\mathbf{p}_{\mathbf{i\_GPS}} = [\mathrm{B}_{\mathrm{long}} \quad \mathrm{B}_{\mathrm{lat}} \quad \mathrm{B}_{\mathrm{alt}}][G^m] + \mathbf{p}_{\mathbf{i\_BFR}} \tag{C.5}$$

We obtain four such equations, one for each point, which are shown in Eqs. (C.6) through (C.9).

$$\mathbf{p}_{\mathbf{1\_GPS}} = [\mathrm{B}_{\mathrm{long}} \quad \mathrm{B}_{\mathrm{lat}} \quad \mathrm{B}_{\mathrm{alt}}][G^m] + \mathbf{p}_{\mathbf{1\_BFR}} \tag{C.6}$$

$$\mathbf{p}_{\mathbf{2\_GPS}} = [\mathrm{B}_{\mathrm{long}} \quad \mathrm{B}_{\mathrm{lat}} \quad \mathrm{B}_{\mathrm{alt}}][G^m] + \mathbf{p}_{\mathbf{2\_BFR}} \tag{C.7}$$

$$\mathbf{p}_{\mathbf{3\_GPS}} = [\mathrm{B}_{\mathrm{long}} \quad \mathrm{B}_{\mathrm{lat}} \quad \mathrm{B}_{\mathrm{alt}}][G^m] + \mathbf{p}_{\mathbf{3\_BFR}} \tag{C.8}$$

$$\mathbf{p}_{\mathbf{4\_GPS}} = [\mathrm{B}_{\mathrm{long}} \quad \mathrm{B}_{\mathrm{lat}} \quad \mathrm{B}_{\mathrm{alt}}][G^m] + \mathbf{p}_{\mathbf{4\_BFR}} \tag{C.9}$$

We then eliminate $[\mathrm{B}_{\mathrm{long}} \quad \mathrm{B}_{\mathrm{lat}} \quad \mathrm{B}_{\mathrm{alt}}][G^m]$, through simple subtraction, and arrive at Eq. (C.10).

$$[B]\,[B^m] = [G][G^m] \tag{C.10}$$

where the matrices $[B]$ and $[G]$ are defined by Eq. (C.11) and (C.12) respectively:

$$[B] = \begin{bmatrix} \mathrm{p}_{1\_BFR\_x} - \mathrm{p}_{2\_BFR\_x} & \mathrm{p}_{1\_BFR\_y} - \mathrm{p}_{2\_BFR\_y} & \mathrm{p}_{1\_BFR\_z} - \mathrm{p}_{2\_BFR\_z} \\ \mathrm{p}_{1\_BFR\_x} - \mathrm{p}_{3\_BFR\_x} & \mathrm{p}_{1\_BFR\_y} - \mathrm{p}_{3\_BFR\_y} & \mathrm{p}_{1\_BFR\_z} - \mathrm{p}_{3\_BFR\_z} \\ \mathrm{p}_{1\_BFR\_x} - \mathrm{p}_{4\_BFR\_x} & \mathrm{p}_{1\_BFR\_y} - \mathrm{p}_{4\_BFR\_y} & \mathrm{p}_{1\_BFR\_z} - \mathrm{p}_{4\_BFR\_z} \end{bmatrix} \tag{C.11}$$

$$[G] = \begin{bmatrix} \mathrm{p}_{1\_GPS\_Long} - \mathrm{p}_{2\_GPS\_Long} & \mathrm{p}_{1\_GPS\_Lat} - \mathrm{p}_{2\_GPS\_Lat} & \mathrm{p}_{1\_GPS\_Alt} - \mathrm{p}_{2\_GPS\_Alt} \\ \mathrm{p}_{1\_GPS\_Long} - \mathrm{p}_{3\_GPS\_Long} & \mathrm{p}_{1\_GPS\_Lat} - \mathrm{p}_{3\_GPS\_Lat} & \mathrm{p}_{1\_GPS\_Alt} - \mathrm{p}_{3\_GPS\_Alt} \\ \mathrm{p}_{1\_GPS\_Long} - \mathrm{p}_{4\_GPS\_Long} & \mathrm{p}_{1\_GPS\_Lat} - \mathrm{p}_{4\_GPS\_Lat} & \mathrm{p}_{1\_GPS\_Alt} - \mathrm{p}_{4\_GPS\_Alt} \end{bmatrix} \tag{C.12}$$

Substituting $[B^m] = [BG][G^m]$ we arrive at Eq. (C.13).

$$[B]\,[BG] = [G] \tag{C.13}$$

Therefore the localized linear transformation matrix $[BG]$ can be computed using Eq. (C.14) as shown below.

$$[BG] = [B]^{-1}[G] \tag{C.14}$$

Using the $[BG]$ matrix, the matrix $[G^m]$ can be expressed in terms of the matrix $[B^m]$. We then determine the GPS coordinates of the origin of the BFR using Eq. (C.15).

$$[\mathrm{B}_{\mathrm{long}} \quad \mathrm{B}_{\mathrm{lat}} \quad \mathrm{B}_{\mathrm{alt}}]\,[G^m] = \mathbf{p}_{\mathbf{1\_GPS}} - \mathbf{p}_{\mathbf{1\_BFR}} \tag{C.15}$$

We then compute the matrix $[I^m]$ in terms of the matrix $[G^m]$ by relating the IFR first with CCF through a rotation transformation, and then relating the CCF with the GPS frame through a scaling transformation. We express the matrix $[I^m]$ as shown in Eq. (C.16).

$$[I^m] = [IG]\,[G^m] \tag{C.16}$$

The orthogonal basis of the GPS frame $[G^m]$ can be represented as $[G^m] = \begin{bmatrix} \mathbf{g}_{\mathbf{long}} \\ \mathbf{g}_{\mathbf{lat}} \\ \mathbf{g}_{\mathbf{alt}} \end{bmatrix}$, where $\mathbf{g}_{\mathbf{long}}$, $\mathbf{g}_{\mathbf{lat}}$, and $\mathbf{g}_{\mathbf{alt}}$ are the unit vectors along the local longitude, latitude and altitude axes. The transformation matrix that scales GPS to CCF is $\begin{bmatrix} 1/\mathbf{g}_{\mathbf{long}} \\ 1/\mathbf{g}_{\mathbf{lat}} \\ 1/\mathbf{g}_{\mathbf{alt}} \end{bmatrix}$. We then compute the [IG] matrix as shown below in Eq. (C.17).

$$[IG] = [R] \begin{bmatrix} 1/\mathbf{g}_{\mathbf{long}} \\ 1/\mathbf{g}_{\mathbf{lat}} \\ 1/\mathbf{g}_{\mathbf{alt}} \end{bmatrix} \tag{C.17}$$

Where, $[R]$ is the rotation transformation matrix that relates IFR and CCF based on the head orientation of the inspector. The calculations establish the values of the $[BG]$ matrix, the $[IG]$ matrix and the GPS coordinates of the origin of the BFR (i.e., $(\mathrm{B}_{\mathrm{long}}, \quad \mathrm{B}_{\mathrm{lat}}, \quad \mathrm{B}_{\mathrm{alt}})$). The algorithm now accepts dynamically changing inspector coordinates in the GPS frame and continuously and accurately registers them to the BFR by using Eq. (C.18), which is derived from Eq. (C.3), Eq. (C.4), Eq. (C.10), and Eq. (C.16).

$$\begin{aligned} \mathbf{p}' = {} & ([\mathrm{P}_{\mathrm{x}} \quad \mathrm{P}_{\mathrm{y}} \quad \mathrm{P}_{\mathrm{z}}]([BG][IG]^{-1})^{-1})[B^m] \\ & +([\mathrm{I}_{\mathrm{long}} - \mathrm{BO}_{\mathrm{long}} \quad \mathrm{I}_{\mathrm{lat}} - \mathrm{BO}_{\mathrm{lat}} \quad \mathrm{I}_{\mathrm{alt}} - \mathrm{BO}_{\mathrm{alt}}]\,[BG]^{-1})[B^m] \end{aligned} \tag{C.18}$$

The above equation is used to register the inspector's location, the near plane center, and the far plane center in the BFR by substituting the following values of $[\mathrm{P}_{\mathrm{x}} \quad \mathrm{P}_{\mathrm{y}} \quad \mathrm{P}_{\mathrm{z}}]$.

- Inspector's location: $\mathrm{P}_{\mathrm{x}} = 0$, $\mathrm{P}_{\mathrm{y}} = 0$, and $\mathrm{P}_{\mathrm{z}} = 0$.
- Near plane center: $\mathrm{P}_{\mathrm{x}} = 0$, P_{y} = near plane distance, and $\mathrm{P}_{\mathrm{z}} = 0$.
- Far plane center: $\mathrm{P}_{\mathrm{x}} = 0$, P_{y} = far plane distance, and $\mathrm{P}_{\mathrm{z}} = 0$.

Thus, we obtain the inspector's location, the near plane center, and far plane center coordinates in BFR.

C.2 Containment Algorithm

The containment algorithm uses the inspector's location coordinates (inspectorX, inspector, inspectorZ), the near plane coordinates (nearX, nearY, nearZ), the far plane coordinates (farX, farY, farZ), the half angle of the cone (viewAngle), and the query point coordinates (pointX, pointY, pointZ) to determine whether or not the query point is in the inspector's field of view. The containment algorithm is implemented through the checkContainment function, as shown in the snippet below.

```
// Containment Algorithm
private static bool checkContainment(double inspectorX, double inspectorY, double
inspectorZ, double nearX, double nearY, double nearZ, double farX, double farY, double
farZ, double viewAngle, double pointX, double pointY, double pointZ)
{
    // Get the normalized line of sight vector
    double[] lineOfSightArray = getLineOfSightVector(farX, farY, farZ, inspectorX,
inspectorY, inspectorZ);

    // Condition 1
    // Check whether the angle between the line of sight vector and the query vector
(joining the inspector to the query point) is less than 90 degrees
    if (((((pointX - inspectorX) * lineOfSightArray[0]) + ((pointY - inspectorY) *
lineOfSightArray[1])) + ((pointZ - inspectorZ) * lineOfSightArray[2])) > 0.0)
    {
         // Get cosine of the angle between the line of sight vector and query vector

         double cosTheta = ((((pointX - inspectorX) * (nearX - inspectorX)) + ((pointY
         - inspectorY) * (nearY - inspectorY))) + ((pointZ - inspectorZ) * (nearZ –
         inspectorZ))) / (normF(pointX - inspectorX, pointY - inspectorY, pointZ –
         inspectorZ) * normF(nearX - inspectorX, nearY - inspectorY, nearZ –
         inspectorZ));

         // Get query angle (between the line of sight vector and the query vector)

         double Theta = Math.Acos(cosTheta);

         if (Theta > 1.5707963267948966)
         {
              Theta = 3.1415926535897931 - Theta;
         }
```

```
          // Condition 2
          // Check whether query angle is less than the inspector's field of view angle
          if (Theta < viewAngle)
          {
               double projectedDistance = normF(inspectorX - pointX, inspectorY –
               pointY, inspectorZ - pointZ) * Math.Cos(Theta);

               // Condition 3
               // Check whether the distance between the inspector and the query point,
               // as projected on the line of sight lies between the near plane
               // distance and the far plane distance
               if((normF(inspectorX - nearX, inspectorY - nearY, inspectorZ - nearZ) <=
                   projectedDistance) && (projectedDistance <= normF(inspectorX - farX,
                   inspectorY - farY, inspectorZ - farZ)))
               {
                    return true; // Return posiive for containment
               }
               else
               {
                    return false;
               }
           }
           else
           {
                return false;
           }
     }
     else
     {
           return false;
     }
}

private static double[] getLineOfSightVector(double FX, double FY, double FZ, double
    IX, double IY, double IZ)
{
    // Get the line of sight vector starting at (IX, IY, IZ) and pointing to (FX, FY,
    // FZ). The vector is (FX - IX)i + (FY - IY)j + (FZ - IZ)k, where i, j, and k are
    // unit vectors of the X, Y, and Z axes

    return new double[] { ((FX - IX) / normF(FX - IX, FY - IY, FZ - IZ)), ((FY - IY) /
        normF(FX - IX, FY - IY, FZ - IZ)), ((FZ - IZ) / normF(FX - IX, FY - IY, FZ –
        IZ)) };
}

private static double normF(double A, double B, double C)
{
    return Math.Sqrt(((A * A) + (B * B)) + (C * C));
}
```

The containment algorithm first determines the normalized line of sight vector (IF in Figure 6.5) using the getLineOfSightVector and normF functions, as shown in the snippet above. For the reasons mentioned in Section 6.3.6, the containment algorithm checks for the following three conditions to determine whether a query point lies in the inspector's field of view.

Condition 1: The algorithm checks whether the angle between the line of sight vector and the query vector (IP in Figure 6.5) is less than 90^o by ensuring that the dot product between these vectors is positive.

Condition 2: The dot product of the normalized line of sight and query vectors is used to estimate the cosine, and thus the angle between these vectors. The algorithm checks whether the angle between the line of sight and query vectors is less than the field of view angle to ensure that query point is not outside the inspector's angular field of view.

Condition 3: The algorithm checks whether the distance between the inspector and the query point as projected onto the line of sight (IP' in Figure 6.5) is more than the inspector's near plane distance (IN in Figure 6.5) and less than the far plane distance (IF in Figure 6.5).

The query point is considered to be in the inspector's spatial-context if, and only if, all three conditions mentioned above are met. If the query point does not meet any of the aforementioned conditions, its location is outside the inspector's truncated viewing cone and is thus not considered to be of contextual relevance.

C.3 Prioritization Algorithm

The containment algorithm is used to process the entire pointcloud to determine contextual points. The points in the pointcloud are grouped according to their corresponding components, and appropriate statistics are collected using the Hashtable data structures with the component as the unique key. As shown in the code snippet below, the following hashtables are used to compile statistics for the prioritization algorithm: 1) NoOfContextualPoints, 2) ClosestDistanceToLineOfSight, and 3) ClosestPointProjectedDistance.

```
// (Contextual Component, # of Contextual Points of that Component)
Hashtable NoOfContextualPoints = new Hashtable();

// (Contextual Component, Closest Point Distance to Line of Sight)
Hashtable ClosestDistanceToLineOfSight = new Hashtable();

// (Contextual Component, Closest Project Distance on the Line of Sight)
Hashtable ClosestPointProjectedDistance = new Hashtable();
```

For each point in the inspector's spatial-context, the algorithm determines: 1) the point's distance from the line of sight (pointDistanceToLineOfSight) (PP' in Figure 6.5), and 2) the distance between the point and the inspector as projected onto the line of sight (projectedDistanceOnLineOfSight) (IP' in Figure 6.5).

```
for (int i = 0; i < pointcloudID.Length; i++) // Process the entire pointcloud
    {

        // Check if point is contextual using containment algorithm
        if (checkContainment(inspectorX, inspectorY, inspectorZ, nearX, nearY, nearZ,
        farX, farY, farZ, viewAngle, pointcloudX[i], pointcloudY[i], pointcloudZ[i]))
        {
            // point distance to line of sight
            double pointDistanceToLineOfSight = pointPdistanceToLineAB(nearX, nearY,
            nearZ, farX, farY, farZ, pointcloudX[i], pointcloudY[i], pointcloudZ[i]);

            // point distance from the inspector
            double pointDistanceToInspector = normF(inspectorX - pointcloudX[i],
            inspectorY - pointcloudY[i], inspectorZ - pointcloudZ[i]);

            // point distance from the inspector as projected onto the line of sight
            double projectedDistanceOnLineOfSight = Math.Sqrt((pointDistanceToInspector
            * pointDistanceToInspector) - (pointDistanceToLineOfSight *
            pointDistanceToLineOfSight));

            // Check whether the component is newly identified as contextual
```

The point's distance to the line of sight is determined using the pointPdistanceToLineAB function. The function determines the distance between a point (Px, Py, Pz) and a line passing through two coordinates ((Ax, Ay, Az) and (Bx, By, Bz)), as shown in the code snippet below.

```
// distance between a point P and a line segment defined by points A and B

private static double pointPdistanceToLineAB(double Ax, double Ay, double Az, double Bx,
    double By, double Bz, double Px, double Py, double Pz)
{
    double slope = ((((Px - Ax) * (Bx - Ax)) + ((Py - Ay) * (By - Ay))) + ((Pz - Az)
    * (Bz - Az))) / ((((Bx - Ax) * (Bx - Ax)) + ((By - Ay) * (By - Ay))) + ((Bz –
    Az) * (Bz - Az)));

    // Coordinates of the point P projected onto the line segment AB
    double projectionX = Ax + (slope * (Bx - Ax));
    double projectionY = Ay + (slope * (By - Ay));
    double projectionZ = Az + (slope * (Bz - Az));

    return Math.Sqrt((((projectionX - Px) * (projectionX - Px)) + ((projectionY - Py) *
    (projectionY - Py))) + ((projectionZ - Pz) * (projectionZ - Pz)));
}
```

After the algorithm computes the values for the pointDistanceToLineOfSight and projectedDistanceOnLineOfSight variables, it checks whether or not the component corresponding to the point was previously identified as being contextual. If the component is being identified as contextual for the first time, a new entry is added to the hashtables using the newly identified component as they unique key, as shown in the code snippet below. The NoOfContextualPoints table is initialized to 1, the pointDistanceToLineOfSight is used for the ClosestDistanceToLineOfSight, and the projectedDistanceOnLineOfSight is used for the ClosestPointProjectedDistance hashtables, as shown below.

```
            // Check whether the component is newly identified as contextual
            if (NoOfContextualPoints.ContainsKey(pointcloudElement[i]))
               {
                   // Component was previously identified as contextual
                   #region
               }
               else
               {
                   // Component is identified as contextual for the first time

                   // Add component to NoOfContextualPoints hashtable
                   NoOfContextualPoints.Add(pointcloudElement[i], 1);

                   // Add component to ClosestDistanceToLineOfSight hashtable
                   ClosestDistanceToLineOfSight.Add(pointcloudElement[i],
                                                      pointDistanceToLineOfSight);

                   // Add component to ClosestPointProjectedDistance hashtable
                   ClosestPointProjectedDistance.Add(pointcloudElement[i],
                                                  projectedDistanceOnLineOfSight);

               }
```

If the component was previously identified as contextual, the hashtable entries are re-assessed to account for the newly identified contextual point, as shown in the code snippet below. The count of the contextual points, as recorded in the NoOfContextualPoints hashtable, is increased by 1. If the newly identified contextual point is closer to the line of sight than all previously identified contextual points (corresponding to the same component), then the newly identified contextual point's pointDistanceToLineOfSight and projectedDistanceOnLineOfSight variables are used for the ClosestDistanceToLineOfSight and ClosestPointProjectedDistance hashtables, respectively.

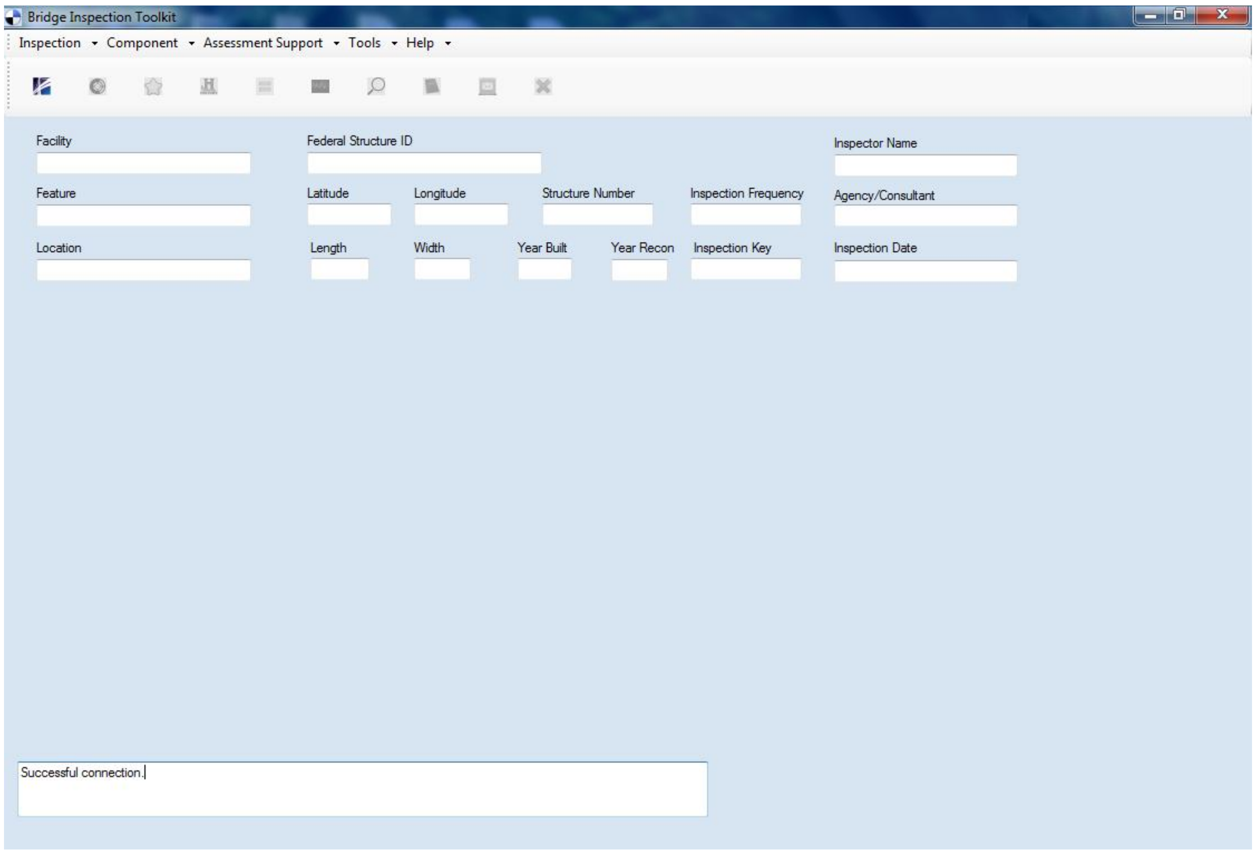
```
// Component was previously identified as contextual
#region

// Increase the count of NoOfContextualPoints by 1

int contextualPointsCount =
                (int)NoOfContextualPoints[pointcloudElement[i]];

NoOfContextualPoints.Remove(pointcloudElement[i]);
     NoOfContextualPoints.Add(pointcloudElement[i],
                                    contextualPointsCount + 1);

// If newly identified point is closer to the line of sight

if (pointDistanceToLineOfSight <
     ((double)ClosestDistanceToLineOfSight[pointcloudElement[i]]))
{
     // Use this point to update ClosestDistanceToLineOfSight

     ClosestDistanceToLineOfSight.Remove(pointcloudElement[i]);

     ClosestDistanceToLineOfSight.Add(pointcloudElement[i],
                                  pointDistanceToLineOfSight);

     // Use this point to update ClosestPointProjected

     ClosestPointProjectedDistance.Remove(pointcloudElement[i]);

     ClosestPointProjectedDistance.Add(pointcloudElement[i],
                               projectedDistanceOnLineOfSight);
}
```

After processing the entire point cloud, the ClosestDistanceToLineOfSight and ClosestPointProjectedDistance hashtables are used to identify the component of interest. For the reasons mentioned in Section 6.3.7, contextual components are first sorted using the ClosestDistanceToLineOfSight hashtable—such that components closest to the line of sight are given a higher priority than components farther from the line of sight. In the case of a tie, the components are sorted using the ClosestPointProjectedDistance hashtable, such that components nearer to the inspector are given a higher priority than components farther from the inspector. After prioritization, the contextual component with the highest priority is considered to be the component of interest to the inspector.

Appendix D

An Introductory Tutorial for the Bridge Inspection Toolkit (BIT)

The Bridge Inspection Toolkit (BIT) is a soft real-time application that allows inspectors to access and record condition assessment information while providing condition assessment decision-making support. The BIT is written in the C# programming language using the WinForms API included as a part of Microsoft's .NET Framework and is implemented to run under the Windows operating system. The BIT communicates with the Central Data Repository (CDR) using the Internet Communications Engine (Ice – 3.4.2) middleware. This appendix explains the procedure for using the BIT. The BIT is designed as a graphical user interface whose screenshot is shown in Figure D.1. Each of the following sections in Appendix D describes the usage of a specific aspect of the BIT.

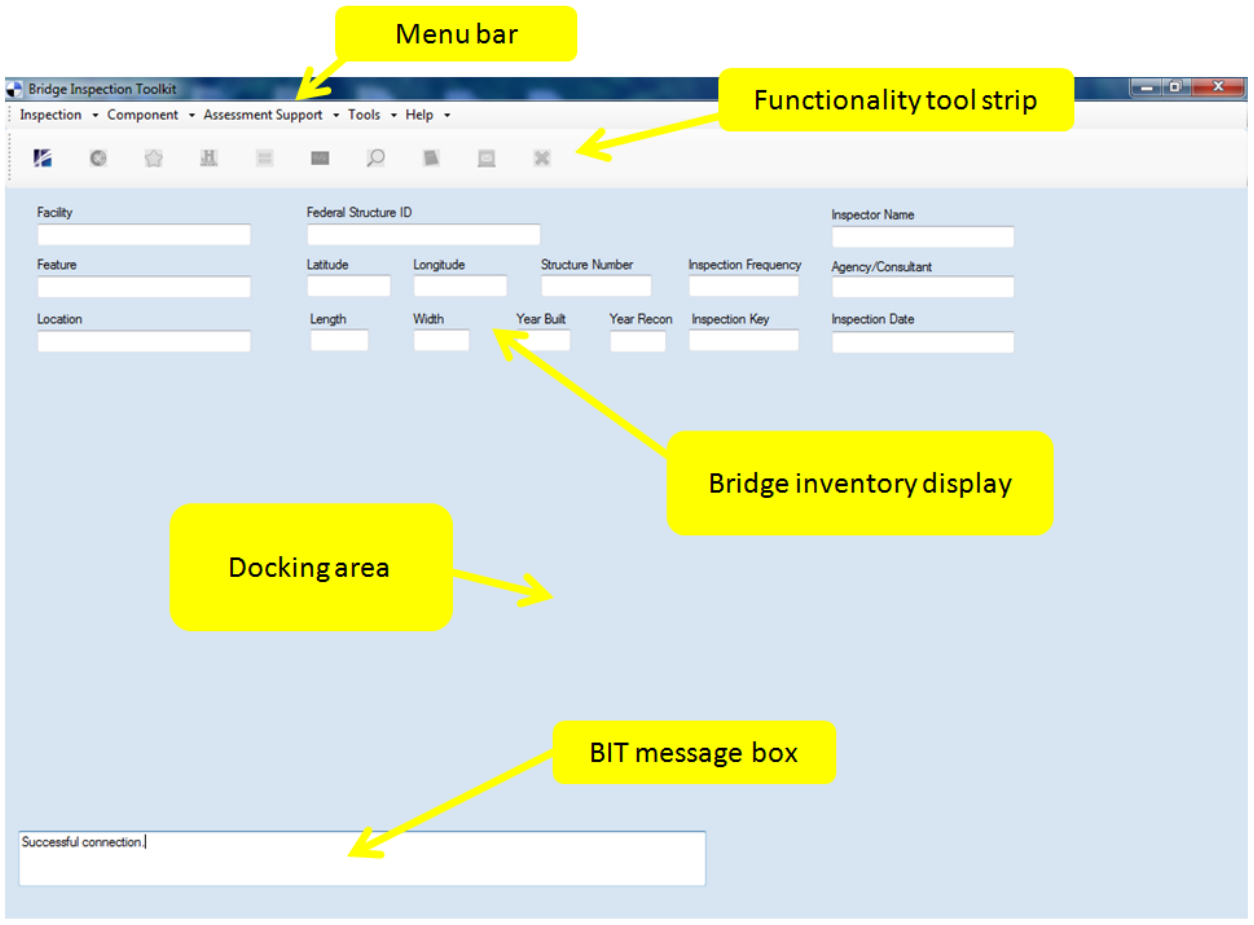


Successful connection.

Figure D.1: The BIT user interface at application startup

D.1 Overview of Menus, Functionalities, Settings, and Support

The BIT is designed and implemented entirely as a graphical user interface as shown in Figure D.2. The top-most area of the interface has a menu bar and a functionality tool strip that allow the user to query the functionality available in the BIT. The area below the functionality tool strip contains a widget that displays inventory information corresponding to the bridge being inspected. The region below this widget is used to dock the BIT functionality widgets. The bottom-most area of the BIT screen contains a message box that communicates the application status to the user.



Successful connection.

Figure D.2: Interface elements of the BIT annotated

The BIT's functionalities—such as component condition rating and Structural Health Monitoring (SHM) analysis—are implemented through independent widgets. These widgets can be accessed by users through their separate interface and commands from the menu bar and the functionality tool strip. Details on using these widgets are described in Sections D.2 through D.6. The basic drop-down user menu bar and the functionality tool strip for the BIT is shown in Figure D.3.

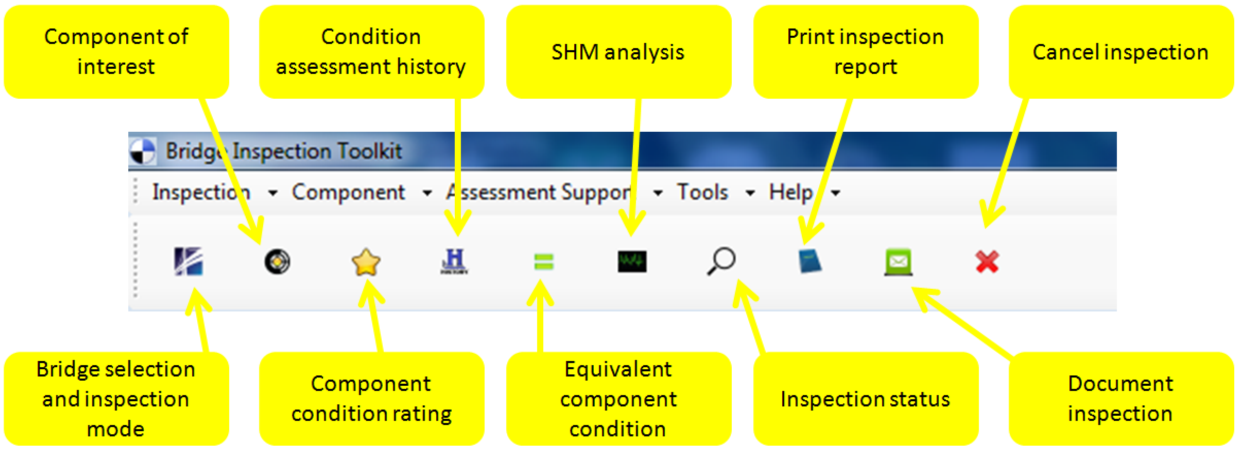
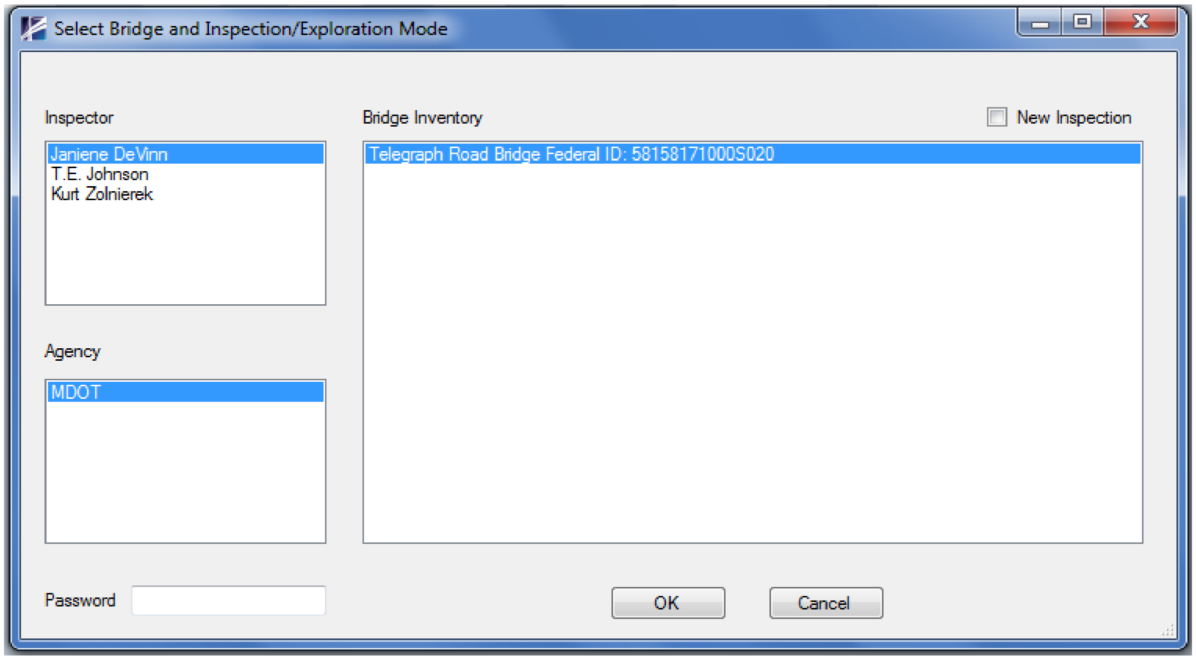


Figure D.3: The BIT user menu and functionality tool strip annotated

D.2 Bridge Selection and Inspection Modes

The user selects the bridge under investigation by querying the bridge selection and inspection mode functionality. Upon query, the user is presented with the bridge selection and inspection mode widget, as shown in Figure D.4. The user enters the appropriate credentials and selects the bridge under investigation from the inventory listed in the widget. By checking (and un-checking) the inspection mode check box, the user can choose to operate the BIT in either 1) the inspection mode or 2) the exploration mode. The widget verifies the user's credentials and returns to the main window of the BIT in the appropriate mode.



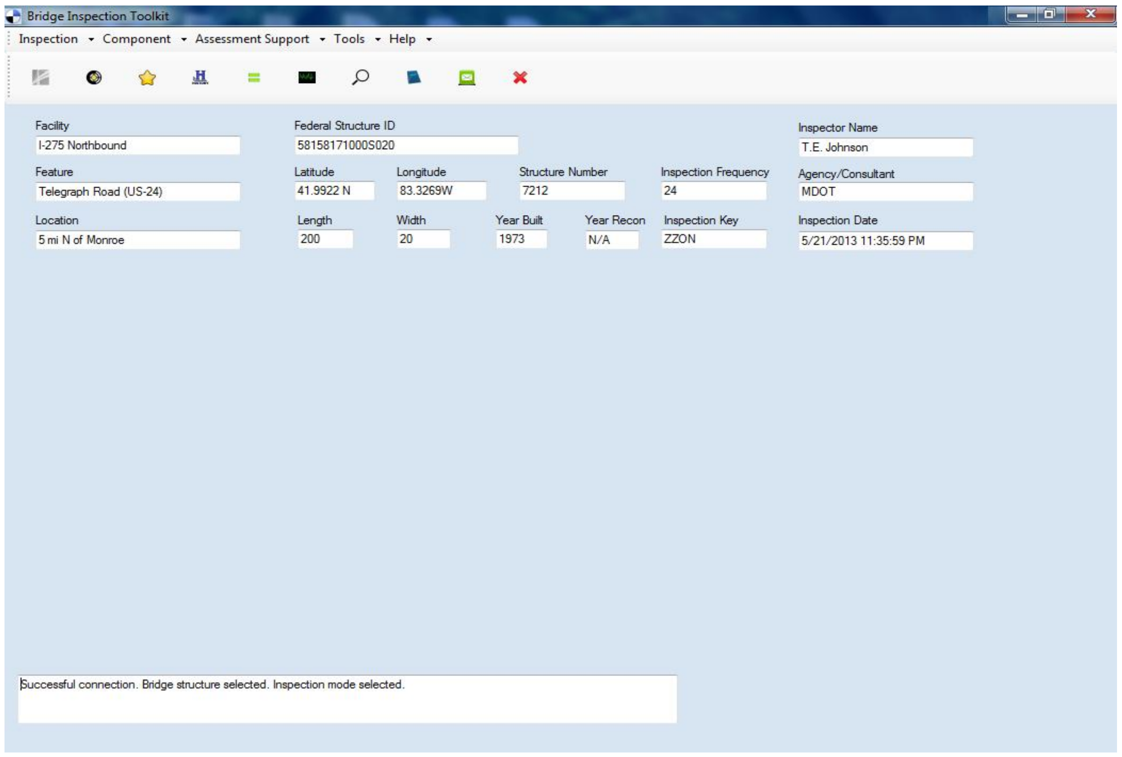
Janiene DeVinn
T.E. Johnson
Kurt Zolnierek

MDOT

Telegraph Road Bridge Federal ID: 58158171000S020

Figure D.4: The bridge selection and inspection mode widget

In the inspection mode, the BIT allows the user to access all functionality available to access and document condition assessment information. The BIT also allows the user to access functionality that provides condition assessment decision-making support. The BIT generates default condition assessment records for all components and inspection-elements corresponding to the selected bridge. In the exploration mode, the BIT allows the user to access previously documented condition assessment information but does not allow the user to record new condition assessment information. Since no new condition assessment information is being documented, the functionality that provides condition assessment decision-making support is unavailable in the exploration mode. The main windows of the BIT in the inspection and exploration modes are shown in Figures D.5 and D.6, respectively.

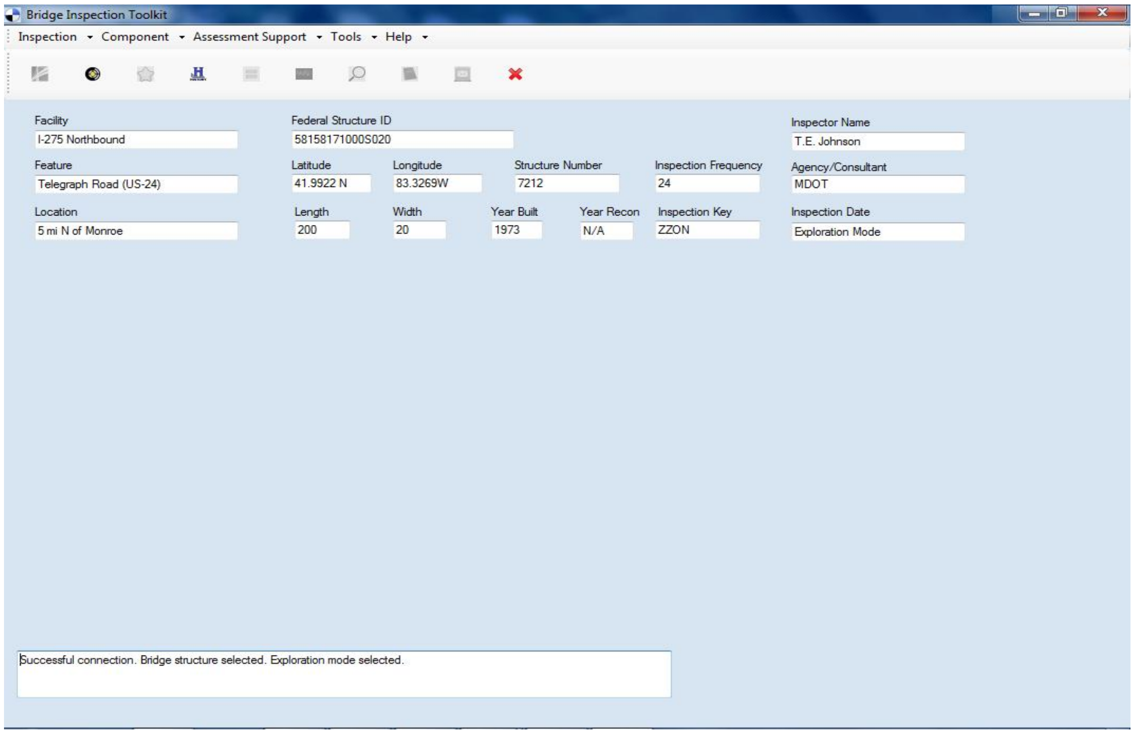


Facility	Federal Structure ID				Inspector Name
I-275 Northbound	58158171000S020				T.E. Johnson
Feature	Latitude	Longitude	Structure Number	Inspection Frequency	Agency/Consultant
Telegraph Road (US-24)	41.9922 N	83.3269W	7212	24	MDOT

Location	Length	Width	Year Built	Year Recon	Inspection Key	Inspection Date
5 mi N of Monroe	200	20	1973	N/A	ZZON	5/21/2013 11:35:59 PM

```
Successful connection. Bridge structure selected. Inspection mode selected.
```

Figure D.5: The main window of the BIT in the inspection mode



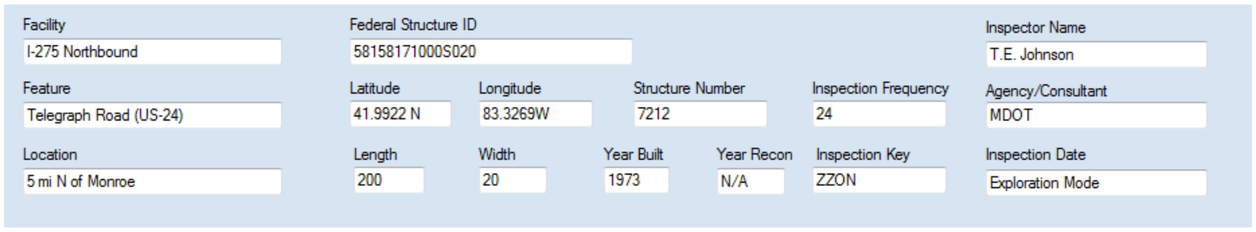
Facility	Federal Structure ID				Inspector Name
I-275 Northbound	58158171000S020				T.E. Johnson
Feature	Latitude	Longitude	Structure Number	Inspection Frequency	Agency/Consultant
Telegraph Road (US-24)	41.9922 N	83.3269W	7212	24	MDOT

Location	Length	Width	Year Built	Year Recon	Inspection Key	Inspection Date
5 mi N of Monroe	200	20	1973	N/A	ZZON	Exploration Mode

```
Successful connection. Bridge structure selected. Exploration mode selected.
```

Figure D.6: The main window of the BIT in the exploration mode

The inventory information corresponding to the selected bridge is displayed in the widget below the functionality tool strip as shown in Figure D.7.

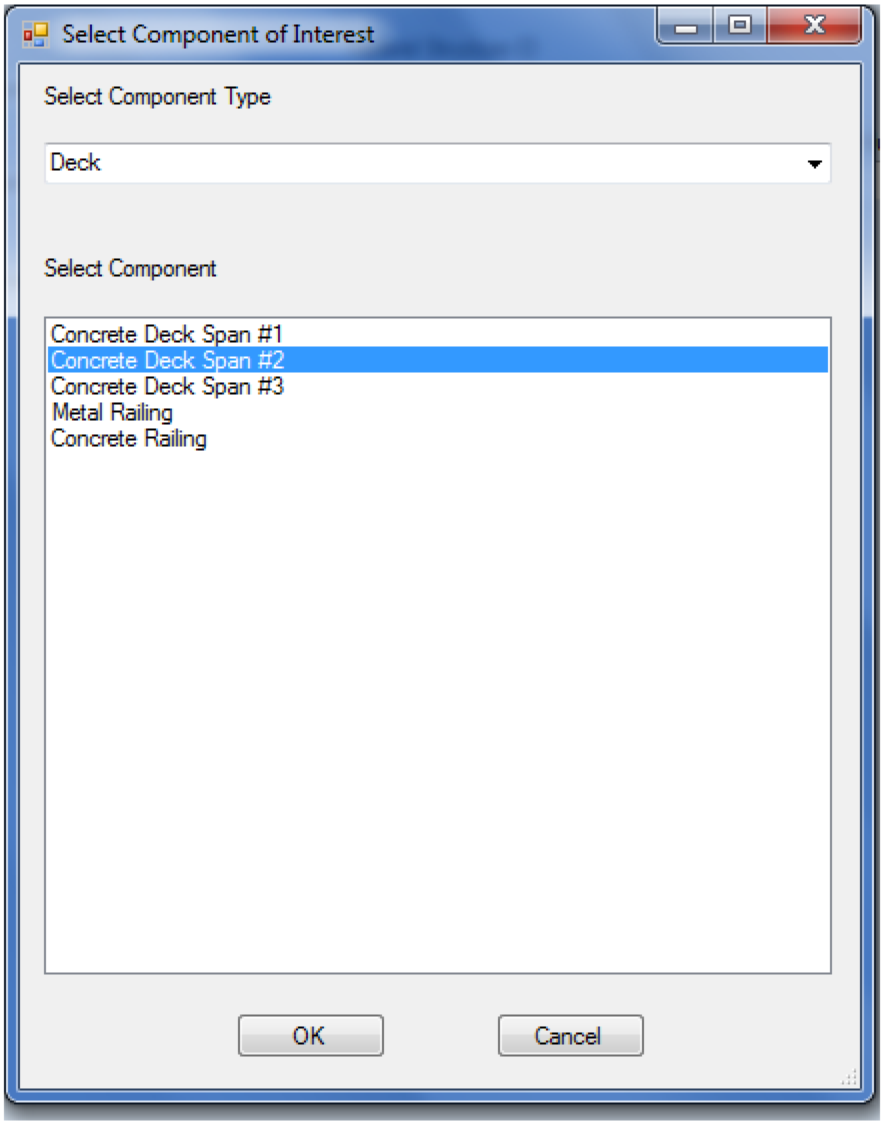


Facility: I-275 Northbound	Federal Structure ID: 58158171000S020					Inspector Name: T.E. Johnson
Feature: Telegraph Road (US-24)	Latitude: 41.9922 N	Longitude: 83.3269W	Structure Number: 7212		Inspection Frequency: 24	Agency/Consultant: MDOT
Location: 5 mi N of Monroe	Length: 200	Width: 20	Year Built: 1973	Year Recon: N/A	Inspection Key: ZZON	Inspection Date: Exploration Mode

Figure D.7: Widget displaying bridge inventory information

D.3 Component of Interest Selection

The component of interest can be selected using either 1) the component of interest widget or 2) a context-aware computing application known as Contextual Object Identifier (CONOID). The user can choose the method for selecting the component of interest by changing the BIT settings. The BIT settings are accessible through the 'Tools' drop-down menu available in the user menu bar. The component of interest widget is shown in Figure D.8. The widget consists of a drop-down combo-box that lists the different types of components corresponding to the bridge under investigation. Upon selection, the widget displays the list of components corresponding to the selected type of component, as shown in Figure D.8. The user then clicks on the component of interest and confirms the selection to the BIT. By changing the BIT settings to acquire the component of interest from CONOID, the user skips the manual selection process described above. The CONOID automatically selects the component of interest for the BIT based on the user's location and head orientation as described in Section 7.2.4.


Select Component of Interest

Select Component Type

Deck

Select Component

Concrete Deck Span #1
Concrete Deck Span #2
Concrete Deck Span #3
Metal Railing
Concrete Railing

Figure D.8: Widget for manually selecting the component of interest

D.4 Condition Assessment History

Upon selecting the component of interest, the user accesses the component's condition assessment data—recorded during prior inspections—by querying the condition assessment history functionality. The user dynamically interacts with the condition assessment history data using the widget shown in Figure D.9.

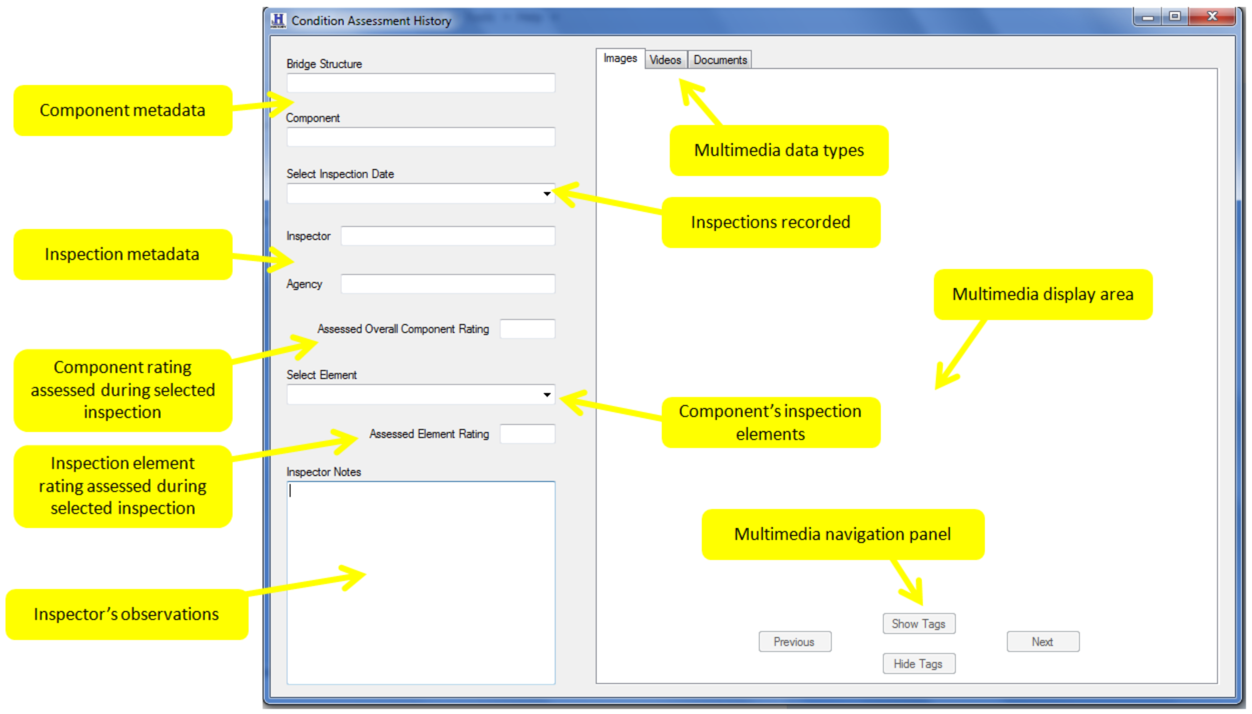
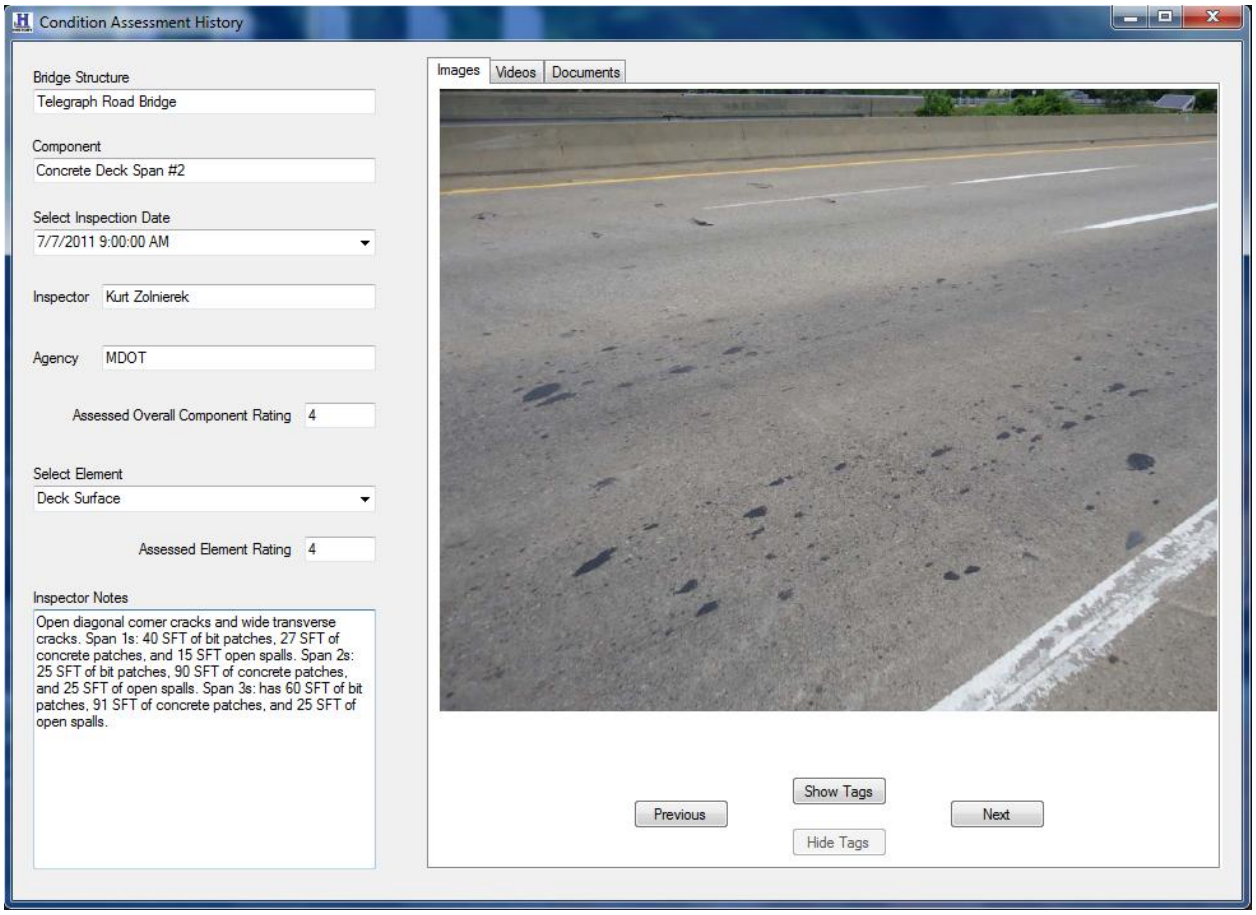


Figure D.9: The component assessment history widget annotated

The left-sided area of the widget displays the component metadata, the list of inspections recorded, the inspection metadata, historic condition assessment ratings, and the inspector's observations. The right-sided area of the widget displays multimedia data—associated with the component—captured during previous inspections. The widget is equipped to handle three types of multimedia data: 1) images, 2) videos, and 3) documents. Upon being invoked, the widget displays the component metadata, the list of inspections recorded previously, and the inspection-elements corresponding to the component of interest. The inspector selects the inspection of interest and the inspection-element of interest using the corresponding drop-down combo-box tools. Upon selection, the widget queries the CDR to retrieve the component's condition rating, the inspection-element's condition rating, and the inspector's observations corresponding to the selected inspection and inspection-element of interest. The widget also retrieves multimedia data—associated with the component—captured during the inspection of interest. The widget displays the retrieved data as shown in Figure D.10. The user browses through multiple multimedia files using the multimedia navigation panel.


Condition Assessment History

Bridge Structure: Telegraph Road Bridge

Component: Concrete Deck Span #2

Select Inspection Date: 7/7/2011 9:00:00 AM

Inspector: Kurt Zolnierek

Agency: MDOT

Assessed Overall Component Rating: 4

Select Element: Deck Surface

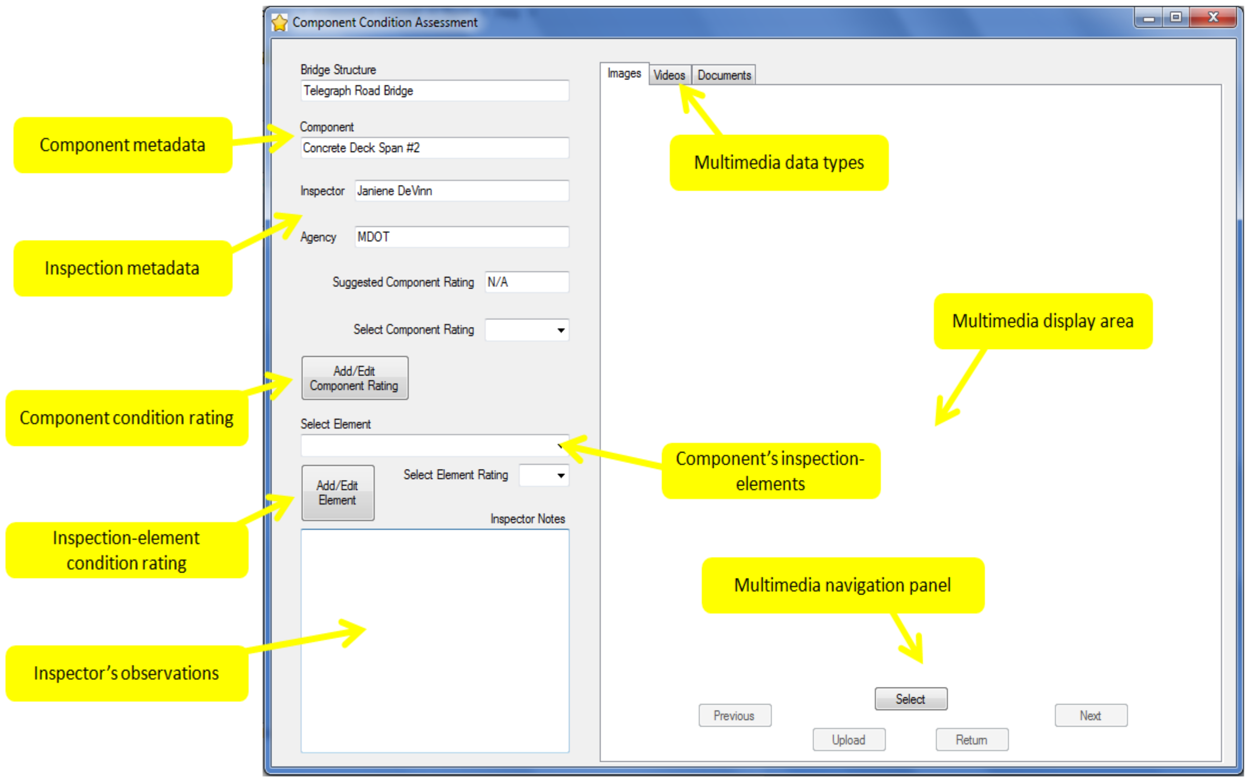
Assessed Element Rating: 4

Inspector Notes: Open diagonal comer cracks and wide transverse cracks. Span 1s: 40 SFT of bit patches, 27 SFT of concrete patches, and 15 SFT open spalls. Span 2s: 25 SFT of bit patches, 90 SFT of concrete patches, and 25 SFT of open spalls. Span 3s: has 60 SFT of bit patches, 91 SFT of concrete patches, and 25 SFT of open spalls.

Figure D.10: The component assessment history widget in action

D.5 Condition Assessment Rating

Upon selecting the component of interest, the user updates the component’s condition assessment and rating data by querying the condition assessment rating functionality. The user dynamically interacts with the condition assessment rating data using the widget shown in Figure D.11.


Component Condition Assessment

Bridge Structure: Telegraph Road Bridge

Component: Concrete Deck Span #2

Inspector: Janiene DeVinn

Agency: MDOT

Suggested Component Rating: N/A

Select Component Rating:

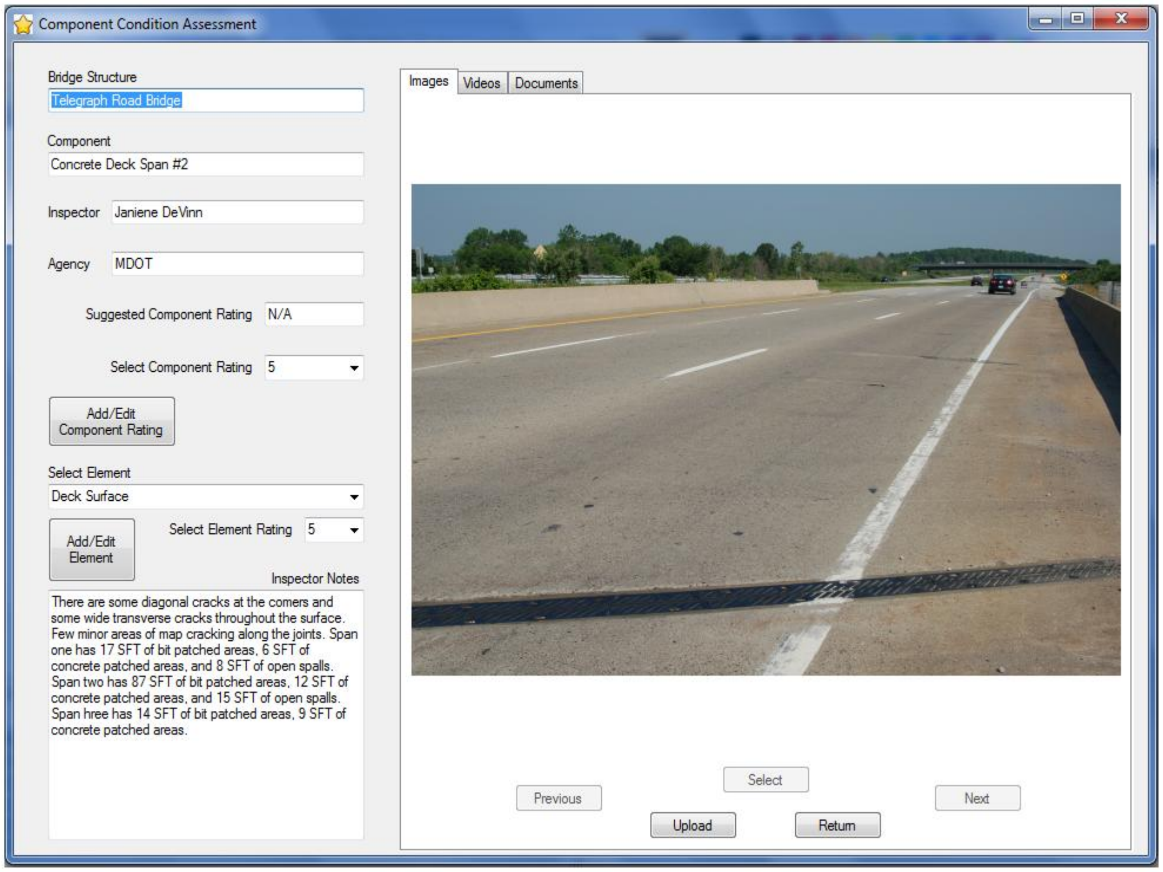
Select Element:

Select Element Rating:

Inspector Notes:

Figure D.11: The condition assessment rating widget annotated

The top-left area of the condition assessment rating widget displays the component and inspection metadata. The bottom-left area of the condition assessment rating widget allows the user to update condition assessment data corresponding to the current inspection. The user selects the inspection-element of interest and assesses its condition by visually inspecting the component. The widget allows the user to update the inspection-element's condition rating and observations, as shown in Figure D.12. Upon updating the inspection-element's condition rating, the widget evaluates the component's condition rating and suggests it to the user, as shown in Figure D.12. The user updates the component's condition rating based on the suggested rating and additional information acquired through the condition assessment decision-making functionality described in Section D.6.


Component Condition Assessment

Bridge Structure: Telegraph Road Bridge

Component: Concrete Deck Span #2

Inspector: Janiene DeVinn

Agency: MDOT

Suggested Component Rating: N/A

Select Component Rating: 5

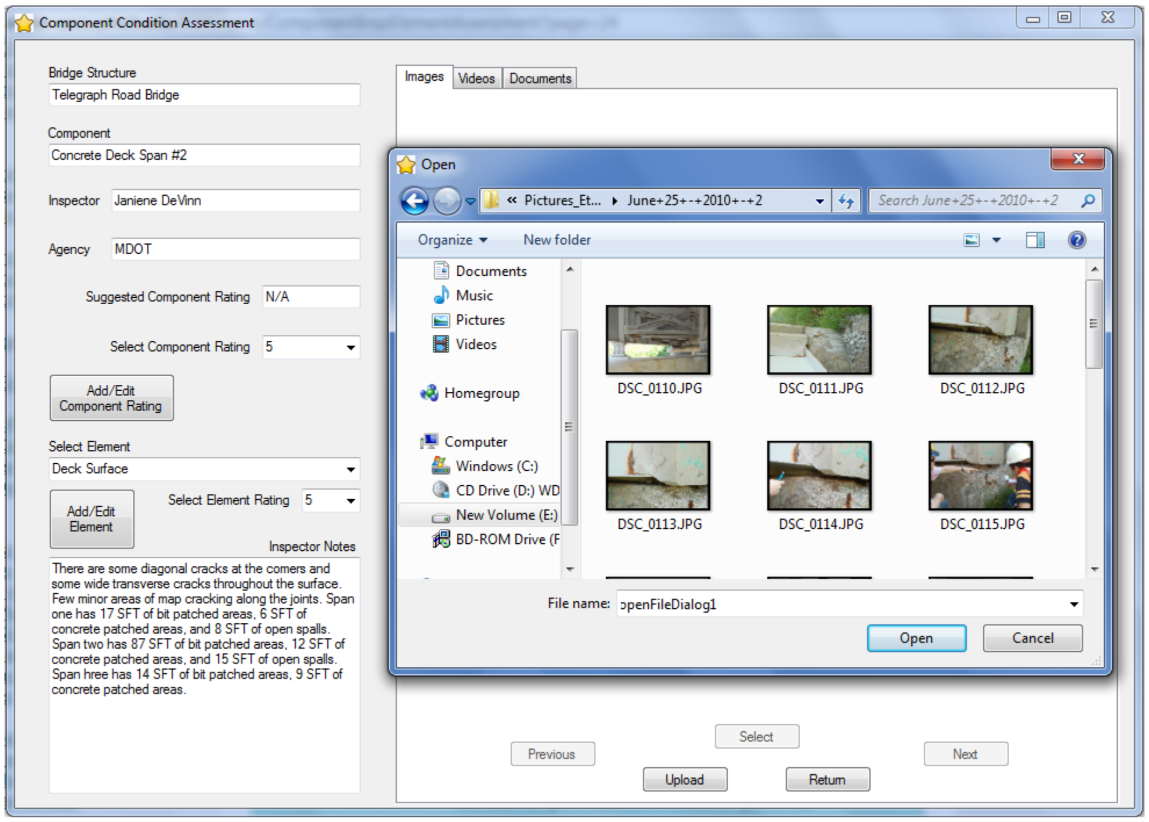
Select Element: Deck Surface

Select Element Rating: 5

Inspector Notes: There are some diagonal cracks at the comers and some wide transverse cracks throughout the surface. Few minor areas of map cracking along the joints. Span one has 17 SFT of bit patched areas, 6 SFT of concrete patched areas, and 8 SFT of open spalls. Span two has 87 SFT of bit patched areas, 12 SFT of concrete patched areas, and 15 SFT of open spalls. Span hree has 14 SFT of bit patched areas, 9 SFT of concrete patched areas.

Figure D.12: The condition assessment rating widget in action

The widget also allows the inspector to associate captured multimedia with the component of interest using the multimedia navigation panel. The widget allows the inspector to select captured multimedia using a file open dialog, as shown in Figure D.13. Upon selection, the user can either 1) upload the selected multimedia to the CDR or 2) discard the selected multimedia. Upon uploading, the widget automatically associates the uploaded multimedia with the component of interest.


Component Condition Assessment

Bridge Structure: Telegraph Road Bridge

Component: Concrete Deck Span #2

Inspector: Janiene DeVinn

Agency: MDOT

Suggested Component Rating: N/A

Select Component Rating: 5

Select Element: Deck Surface

Select Element Rating: 5

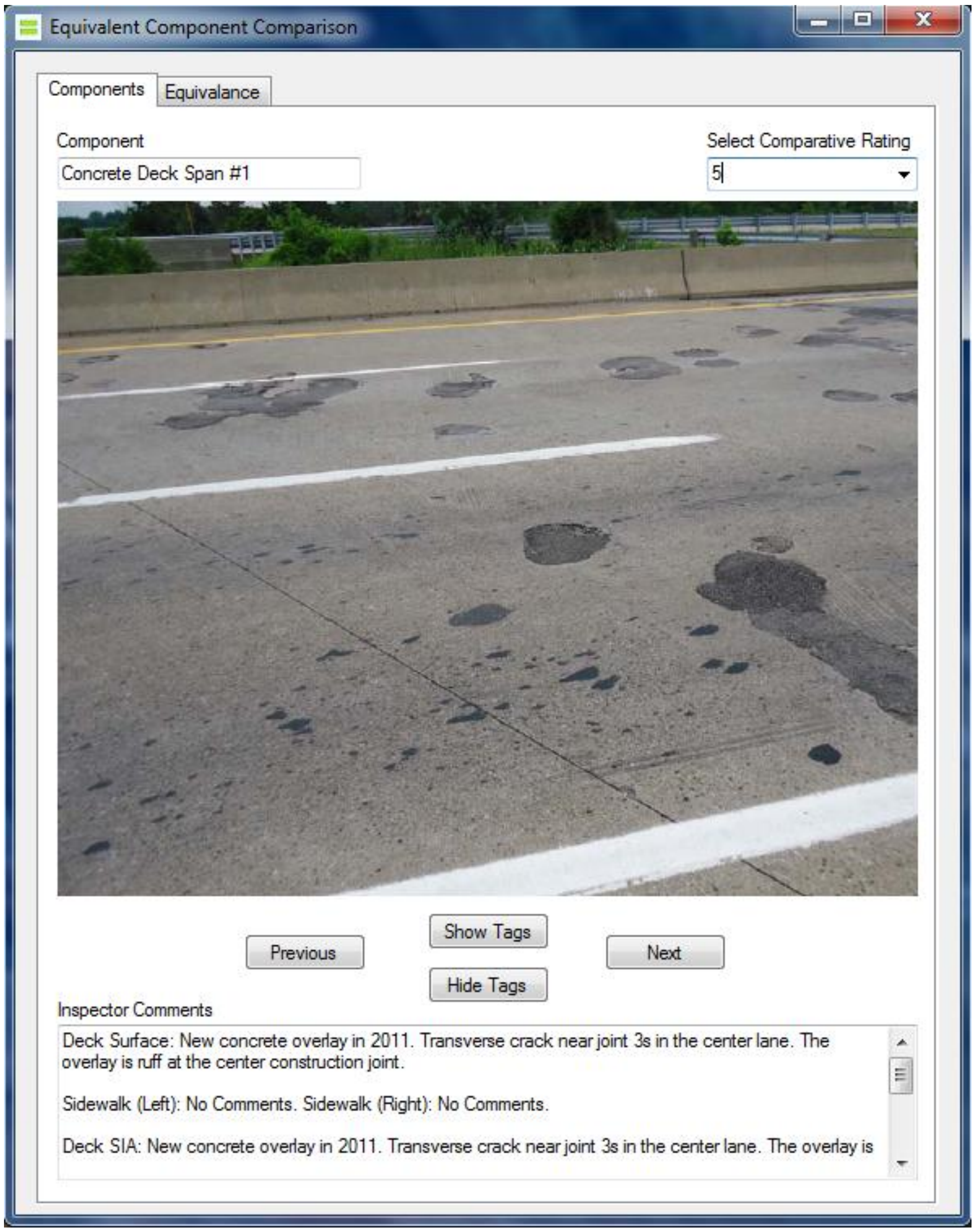
Inspector Notes: There are some diagonal cracks at the comers and some wide transverse cracks throughout the surface. Few minor areas of map cracking along the joints. Span one has 17 SFT of bit patched areas, 6 SFT of concrete patched areas, and 8 SFT of open spalls. Span two has 87 SFT of bit patched areas, 12 SFT of concrete patched areas, and 15 SFT of open spalls. Span hree has 14 SFT of bit patched areas, 9 SFT of concrete patched areas.

File name: openFileDialog1

Figure D.13: File open dialog for selecting captured multimedia files

D.6 Condition Assessment Decision-Making Support

Condition assessment decision-making support is provided using two functionalities: 1) equivalent component comparison, and 2) SHM analysis. Equivalent component comparison functionality allows the user to draw on the experience and judgment of peers by comparing the condition of the component in context with equivalent components across the rating scale. Upon query, the user is directed to the equivalent component comparison widget, shown in Figure D.14, which provides the ability required for comparing equivalent components. The user selects the condition rating deemed appropriate for the component of interest using the dropdown box. Upon selection, the widget queries the CDR to retrieve assessment information for similar components belonging to equivalent bridges with the same condition rating. The user views captured multimedia, inspection notes, and observations associated with similarly rated equivalent components in the inventory. The user refines the component's condition rating by browsing for condition concurrence among equivalent components.



Component: Concrete Deck Span #1

Select Comparative Rating: 5

Inspector Comments

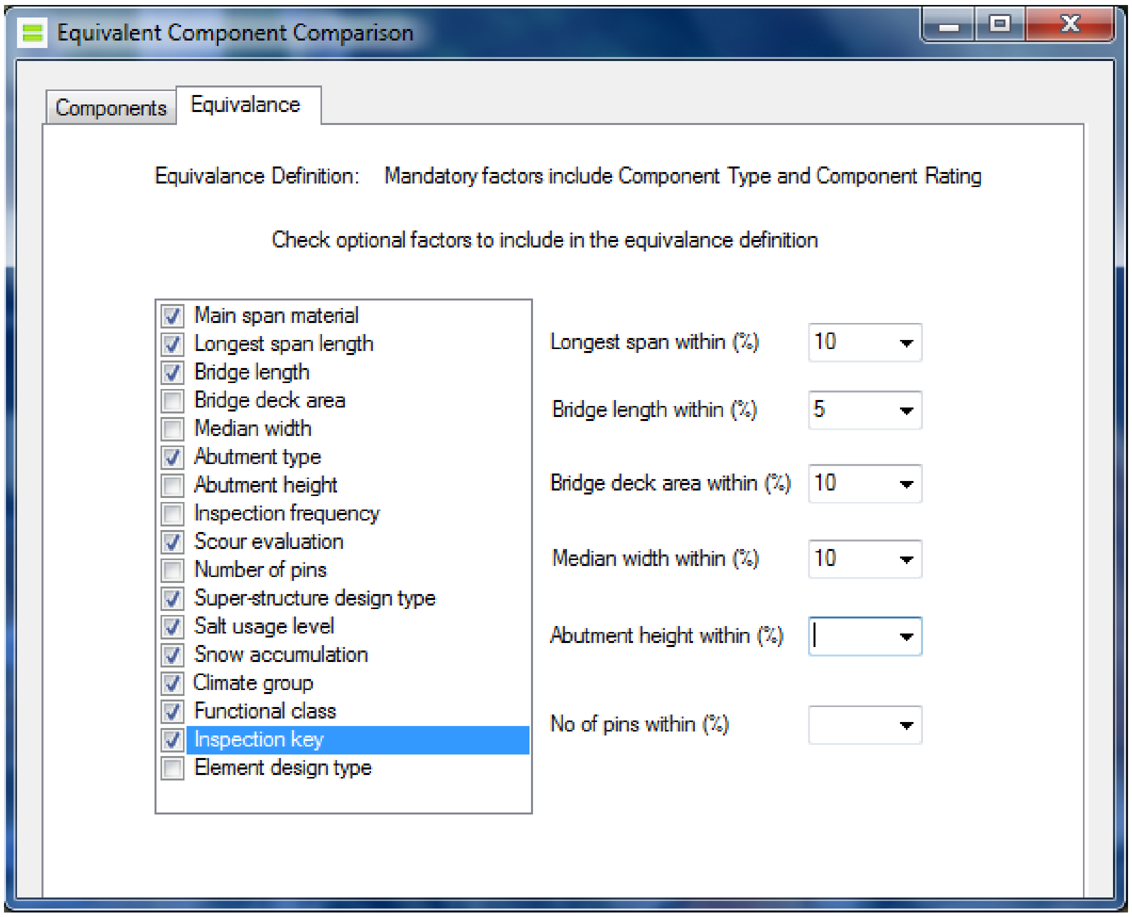
Deck Surface: New concrete overlay in 2011. Transverse crack near joint 3s in the center lane. The overlay is ruff at the center construction joint.

Sidewalk (Left): No Comments. Sidewalk (Right): No Comments.

Deck SIA: New concrete overlay in 2011. Transverse crack near joint 3s in the center lane. The overlay is

Figure D.14: The equivalent component comparison widget in action

The equivalent component comparison widget also allows the user to define similarity (equivalence) among bridges by switching to the equivalence definition tab, as shown in Figure D.15. Equivalence between components is defined based on bridge and traffic characteristics.



Equivalance Definition: Mandatory factors include Component Type and Component Rating

Check optional factors to include in the equivalance definition

- [✓] Main span material
- [✓] Longest span length
- [✓] Bridge length
- [ ] Bridge deck area
- [ ] Median width
- [✓] Abutment type
- [ ] Abutment height
- [ ] Inspection frequency
- [✓] Scour evaluation
- [ ] Number of pins
- [✓] Super-structure design type
- [✓] Salt usage level
- [✓] Snow accumulation
- [✓] Climate group
- [✓] Functional class
- [✓] Inspection key
- [ ] Element design type

Longest span within (%)	10
Bridge length within (%)	5
Bridge deck area within (%)	10
Median width within (%)	10
Abutment height within (%)	
No of pins within (%)	

Figure D.15: The equivalence definition tab of the equivalent component comparison widget

The second functionality employed by the BIT to provide condition assessment decision-making support is the SHM analysis. Upon query, the user is directed to the web-based sensor analytics form, as shown in Figure D.16. The URL of the web-form is customized based on the component of interest. The web-form has URL links to raw and processed SHM information corresponding to the component. The user requests the desired analysis by clicking on the corresponding URL to invoke appropriate Python scripts in the sensor analytics client. The sensor analytics client runs the SHM algorithms and returns the resulting Google Chart object as an output. The interactive Google Chart object is embedded into the web-browser and presented to the user. The user then interprets the presented SHM analysis and refines the component's condition rating.

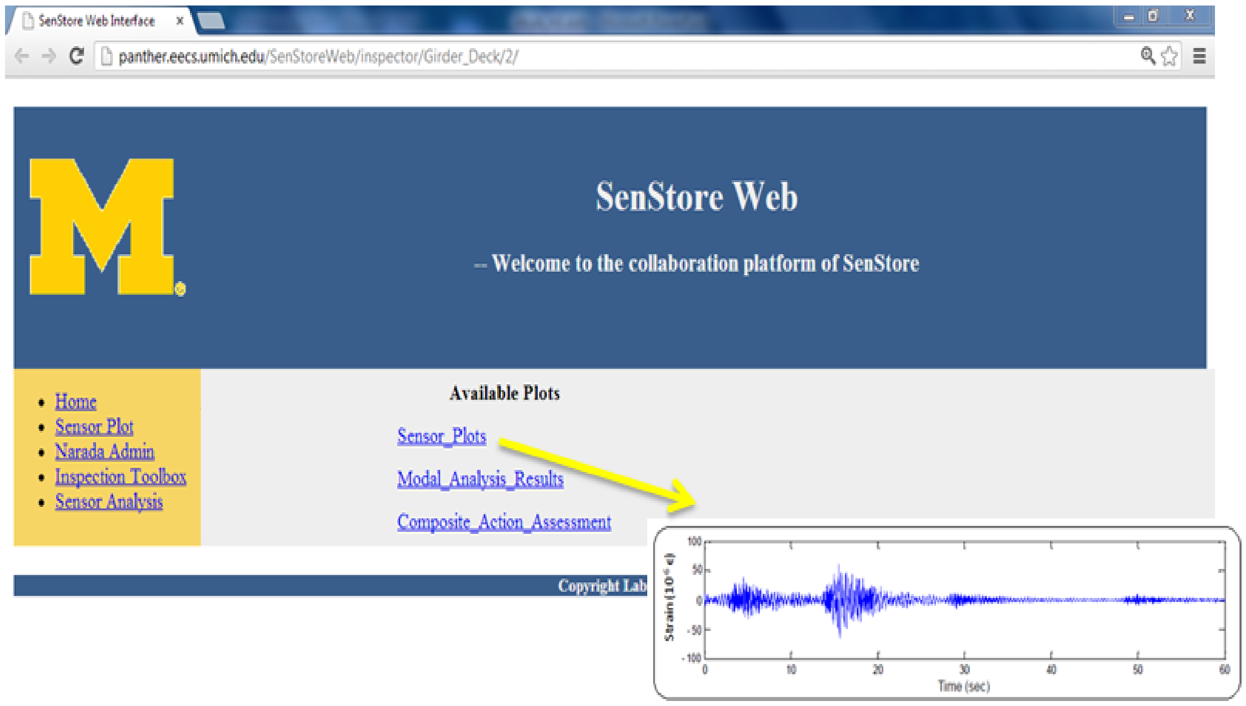
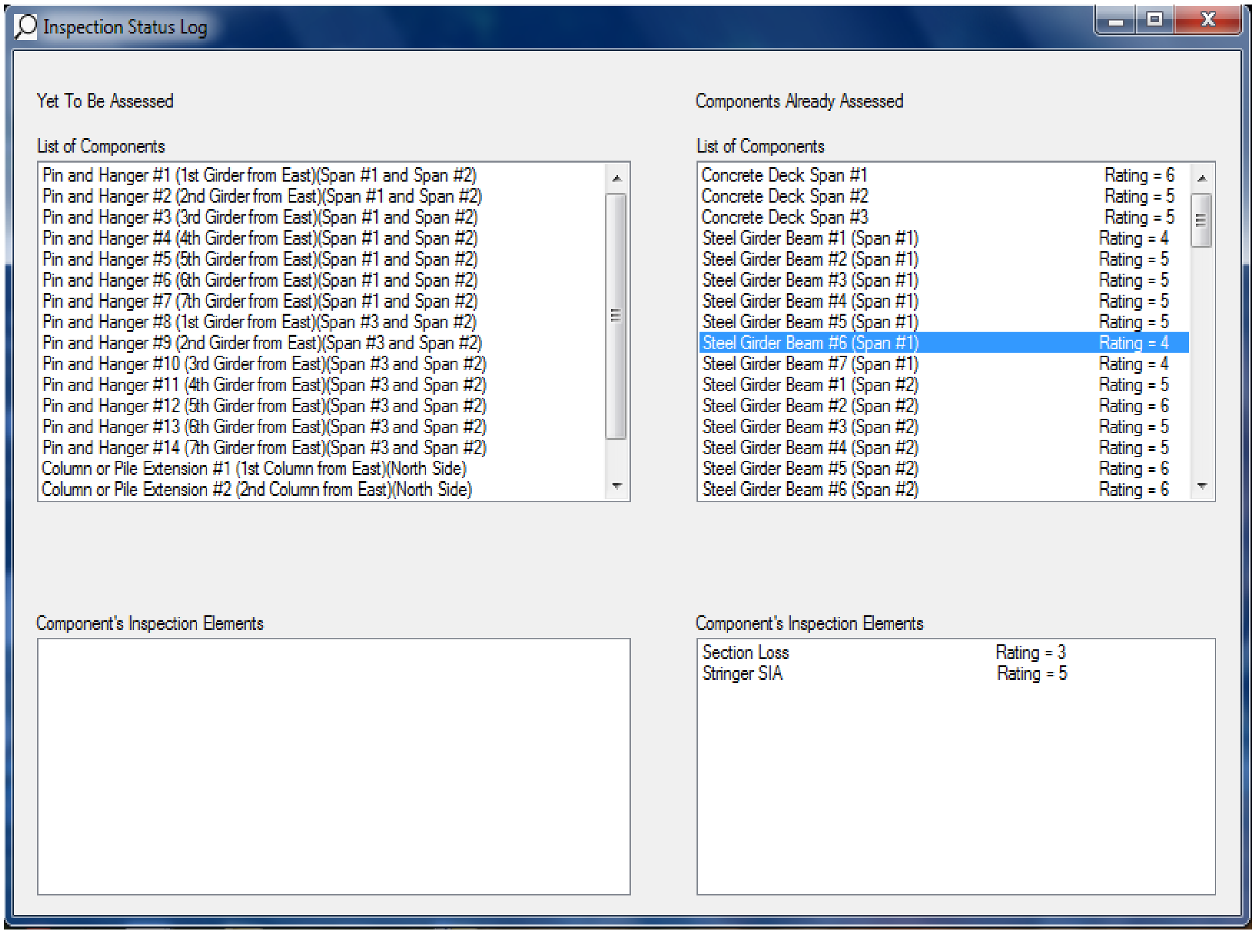


Figure D.16: The sensor analytics web-form generated by the SHM analysis functionality

D.7 Inspection Support Tools

The BIT allows the user to track the progress of the inspection operation by keeping a log of those components and inspection-elements that have been assessed and those that are yet to be assessed. The inspection progress log is accessed using the inspection status functionality available in the main window of the BIT. Figure D.17 shows the inspection progress log captured during a typical inspection routine.



Yet To Be Assessed

List of Components

Pin and Hanger #1 (1st Girder from East)(Span #1 and Span #2)
Pin and Hanger #2 (2nd Girder from East)(Span #1 and Span #2)
Pin and Hanger #3 (3rd Girder from East)(Span #1 and Span #2)
Pin and Hanger #4 (4th Girder from East)(Span #1 and Span #2)
Pin and Hanger #5 (5th Girder from East)(Span #1 and Span #2)
Pin and Hanger #6 (6th Girder from East)(Span #1 and Span #2)
Pin and Hanger #7 (7th Girder from East)(Span #1 and Span #2)
Pin and Hanger #8 (1st Girder from East)(Span #3 and Span #2)
Pin and Hanger #9 (2nd Girder from East)(Span #3 and Span #2)
Pin and Hanger #10 (3rd Girder from East)(Span #3 and Span #2)
Pin and Hanger #11 (4th Girder from East)(Span #3 and Span #2)
Pin and Hanger #12 (5th Girder from East)(Span #3 and Span #2)
Pin and Hanger #13 (6th Girder from East)(Span #3 and Span #2)
Pin and Hanger #14 (7th Girder from East)(Span #3 and Span #2)
Column or Pile Extension #1 (1st Column from East)(North Side)
Column or Pile Extension #2 (2nd Column from East)(North Side)

Component's Inspection Elements

Components Already Assessed

List of Components

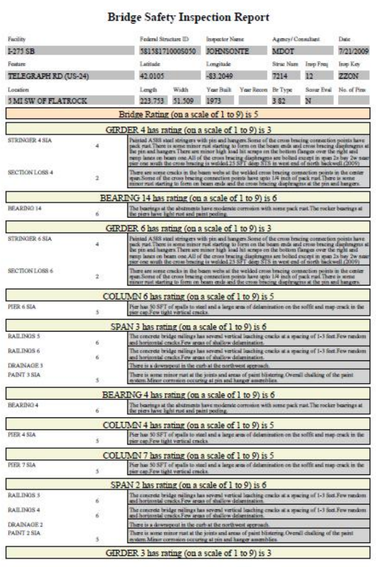
Concrete Deck Span #1	Rating = 6
Concrete Deck Span #2	Rating = 5
Concrete Deck Span #3	Rating = 5
Steel Girder Beam #1 (Span #1)	Rating = 4
Steel Girder Beam #2 (Span #1)	Rating = 5
Steel Girder Beam #3 (Span #1)	Rating = 5
Steel Girder Beam #4 (Span #1)	Rating = 5
Steel Girder Beam #5 (Span #1)	Rating = 5
Steel Girder Beam #6 (Span #1)	Rating = 4
Steel Girder Beam #7 (Span #1)	Rating = 4
Steel Girder Beam #1 (Span #2)	Rating = 5
Steel Girder Beam #2 (Span #2)	Rating = 6
Steel Girder Beam #3 (Span #2)	Rating = 5
Steel Girder Beam #4 (Span #2)	Rating = 5
Steel Girder Beam #5 (Span #2)	Rating = 6
Steel Girder Beam #6 (Span #2)	Rating = 6

Component's Inspection Elements

Section Loss	Rating = 3
Stringer SIA	Rating = 5

Figure D.17: The inspection progress log captured during a typical inspection routine

Upon completing an inspection, the user can generate information-rich inspection reports by invoking the report generation functionality available in the main window of the BIT. The inspection report is customized to the format used by MDOT and is generated as a Portable Document Format (PDF) file. Figure D.18 shows excerpts from the bridge inspection report of an inspection routine conducted on a typical highway bridge.


Bridge Safety Inspection Report

Facility		Federal Structure ID	Inspector Name		Agency/Consultant		Date
I-275 SB		58158171000S050	JOHNSONTE		MDOT		7/21/2009
Feature		Latitude	Longitude		Struc Num	Insp Freq	Insp Key
TELEGRAPH RD (US-24)		42.0105	-83.2049		7214	12	ZZON
Location		Length	Width	Year Built	Year Recon	Br Type	Scour Eval	No. of Pins
5 MI SW OF FLATROCK		223.753	51.509	1973		3 82	N	

Bridge Rating (on a scale of 1 to 9) is 5

GIRDER 4 has rating (on a scale of 1 to 9) is 3

STRINGER 4 SIA	4	Painted A588 steel stringers with pin and hangers.Some of the cross bracing connection points have pack rust.There is some minor rust starting to form on the beam ends and cross bracing diaphragms at the pin and hangers.There are minor high load hit scraps on the bottom flanges over the right and ramp lanes on beam one.All of the cross bracing diaphragms are bolted except in span 2s bay 2w near pier one south the cross bracing is welded.25 SFT deep STS in west end of north backwell (2009)
SECTION LOSS 4	2	There are some cracks in the beam webs at the welded cross bracing connection points in the center span.Some of the cross bracing connection points have upto 1/4 inch of pack rust.There is some minor rust starting to form on beam ends and the cross bracing diaphragms at the pin and hangers.

BEARING 14 has rating (on a scale of 1 to 9) is 6

BEARING 14	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

GIRDER 6 has rating (on a scale of 1 to 9) is 3

STRINGER 6 SIA	4	Painted A588 steel stringers with pin and hangers.Some of the cross bracing connection points have pack rust.There is some minor rust starting to form on the beam ends and cross bracing diaphragms at the pin and hangers.There are minor high load hit scraps on the bottom flanges over the right and ramp lanes on beam one.All of the cross bracing diaphragms are bolted except in span 2s bay 2w near pier one south the cross bracing is welded.25 SFT deep STS in west end of north backwell (2009)
SECTION LOSS 6	2	There are some cracks in the beam webs at the welded cross bracing connection points in the center span.Some of the cross bracing connection points have upto 1/4 inch of pack rust.There is some minor rust starting to form on beam ends and the cross bracing diaphragms at the pin and hangers.

COLUMN 6 has rating (on a scale of 1 to 9) is 5

PIER 6 SIA	5	Pier has 50 SFT of spalls to steel and a large area of delamination on the soffit and map crack in the pier cap.Few tight vertical cracks.

SPAN 3 has rating (on a scale of 1 to 9) is 6

RAILINGS 5	6	The concrete bridge railings has several vertical leaching cracks at a spacing of 1-3 feet.Few random and horizontal cracks.Few areas of shallow delamination.
RAILINGS 6	6	The concrete bridge railings has several vertical leaching cracks at a spacing of 1-3 feet.Few random and horizontal cracks.Few areas of shallow delamination.
DRAINAGE 3		There is a downspout in the curb at the northwest approach.
PAINT 3 SIA	5	There is some minor rust at the joints and areas of paint blistering.Overall chalking of the paint system.Minor corrosion occuring at pin and hanger assemblies.

BEARING 4 has rating (on a scale of 1 to 9) is 6

BEARING 4	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

COLUMN 4 has rating (on a scale of 1 to 9) is 5

PIER 4 SIA	5	Pier has 50 SFT of spalls to steel and a large area of delamination on the soffit and map crack in the pier cap.Few tight vertical cracks.

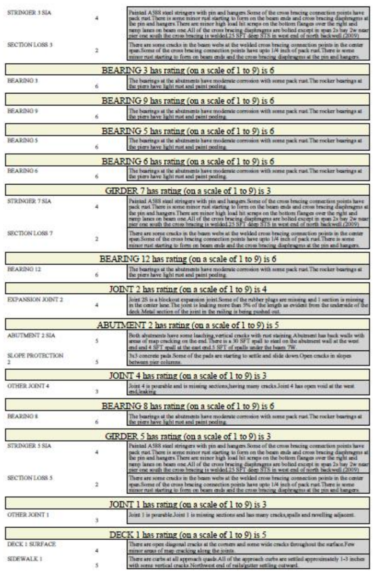
COLUMN 7 has rating (on a scale of 1 to 9) is 5

PIER 7 SIA	5	Pier has 50 SFT of spalls to steel and a large area of delamination on the soffit and map crack in the pier cap.Few tight vertical cracks.

SPAN 2 has rating (on a scale of 1 to 9) is 6

RAILINGS 3	6	The concrete bridge railings has several vertical leaching cracks at a spacing of 1-3 feet.Few random and horizontal cracks.Few areas of shallow delamination.
RAILINGS 4	6	The concrete bridge railings has several vertical leaching cracks at a spacing of 1-3 feet.Few random and horizontal cracks.Few areas of shallow delamination.
DRAINAGE 2		There is a downspout in the curb at the northwest approach.
PAINT 2 SIA	5	There is some minor rust at the joints and areas of paint blistering.Overall chalking of the paint system.Minor corrosion occuring at pin and hanger assemblies.

GIRDER 3 has rating (on a scale of 1 to 9) is 3



STRINGER 3 SIA	4	Painted A588 steel stringers with pin and hangers.Some of the cross bracing connection points have pack rust.There is some minor rust starting to form on the beam ends and cross bracing diaphragms at the pin and hangers.There are minor high load hit scraps on the bottom flanges over the right and ramp lanes on beam one.All of the cross bracing diaphragms are bolted except in span 2s bay 2w near pier one south the cross bracing is welded.25 SFT deep STS in west end of north backwell (2009)
SECTION LOSS 3	2	There are some cracks in the beam webs at the welded cross bracing connection points in the center span.Some of the cross bracing connection points have upto 1/4 inch of pack rust.There is some minor rust starting to form on beam ends and the cross bracing diaphragms at the pin and hangers.

BEARING 3 has rating (on a scale of 1 to 9) is 6

BEARING 3	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

BEARING 9 has rating (on a scale of 1 to 9) is 6

BEARING 9	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

BEARING 5 has rating (on a scale of 1 to 9) is 6

BEARING 5	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

BEARING 6 has rating (on a scale of 1 to 9) is 6

BEARING 6	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

GIRDER 7 has rating (on a scale of 1 to 9) is 3

STRINGER 7 SIA	4	Painted A588 steel stringers with pin and hangers.Some of the cross bracing connection points have pack rust.There is some minor rust starting to form on the beam ends and cross bracing diaphragms at the pin and hangers.There are minor high load hit scraps on the bottom flanges over the right and ramp lanes on beam one.All of the cross bracing diaphragms are bolted except in span 2s bay 2w near pier one south the cross bracing is welded.25 SFT deep STS in west end of north backwell (2009)
SECTION LOSS 7	2	There are some cracks in the beam webs at the welded cross bracing connection points in the center span.Some of the cross bracing connection points have upto 1/4 inch of pack rust.There is some minor rust starting to form on beam ends and the cross bracing diaphragms at the pin and hangers.

BEARING 12 has rating (on a scale of 1 to 9) is 6

BEARING 12	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

JOINT 2 has rating (on a scale of 1 to 9) is 4

EXPANSION JOINT 2	4	Joint 2S is a blockout expansion joint.Some of the rubber plugs are missing and 1 section is missing in the center lane.The joint is leaking more than 5% of the length as evident from the underside of the deck.Metal section of the joint in the railing is being pushed out.

ABUTMENT 2 has rating (on a scale of 1 to 9) is 5

ABUTMENT 2 SIA	5	Both abutments have some leaching,vertical cracks with rust staining.Abutment has back walls with areas of map cracking on the end.There is a 30 SFT spall to steel on the abutment wall at the west end and 4 SFT spall at the east end.5 SFT of spalls under the beam 7W.
SLOPE PROTECTION 2	5	3x3 concrete pads.Some of the pads are starting to settle and slide down.Open cracks in slopes between pier columns.

JOINT 4 has rating (on a scale of 1 to 9) is 3

OTHER JOINT 4	3	Joint 4 is pourable and is missing sections,having many cracks.Joint 4 has open void at the west end,leaking.

BEARING 8 has rating (on a scale of 1 to 9) is 6

BEARING 8	6	The bearings at the abutments have moderate corrosion with some pack rust.The rocker bearings at the piers have light rust and paint peeling.

GIRDER 5 has rating (on a scale of 1 to 9) is 3

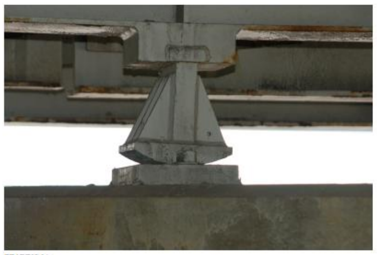
STRINGER 5 SIA	4	Painted A588 steel stringers with pin and hangers.Some of the cross bracing connection points have pack rust.There is some minor rust starting to form on the beam ends and cross bracing diaphragms at the pin and hangers.There are minor high load hit scraps on the bottom flanges over the right and ramp lanes on beam one.All of the cross bracing diaphragms are bolted except in span 2s bay 2w near pier one south the cross bracing is welded.25 SFT deep STS in west end of north backwell (2009)
SECTION LOSS 5	2	There are some cracks in the beam webs at the welded cross bracing connection points in the center span.Some of the cross bracing connection points have upto 1/4 inch of pack rust.There is some minor rust starting to form on beam ends and the cross bracing diaphragms at the pin and hangers.

JOINT 1 has rating (on a scale of 1 to 9) is 3

OTHER JOINT 1	3	Joint 1 is pourable.Joint 1 is missing sections and has many cracks,spalls and ravelling adjacent.

DECK 1 has rating (on a scale of 1 to 9) is 5

DECK 1 SURFACE	4	There are open diagonal cracks at the corners and some wide cracks throughout the surface.Few minor areas of map cracking along the joints.
SIDEWALK 1	5	There are curbs at all approach quads.All of the approach curbs are settled approximately 1-3 inches with some vertical cracks.Northwest end of rails/gutter settling outward.


BEARING 14

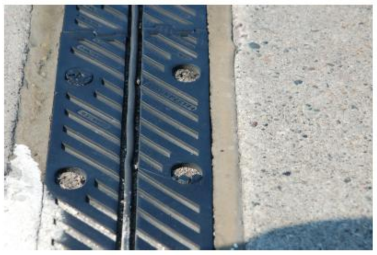

EXPANSION JOINT 2

Figure D.18: Bridge inspection report generated by the BIT

D.8 Registering and/or Canceling Inspection

The user can cancel the inspection being recorded at anytime during the inspection by invoking the cancel inspection functionality. The cancel inspection functionality deletes appropriate data and metadata entries—relevant to the canceled inspection—from the CDR. The user can register a successfully concluded inspection using the document inspection functionality.