Real-Time Visualization for Prevention of Excavation Related Utility Strikes

by

Sanat A. Talmaki

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering)
in The University of Michigan
2012

Doctoral Committee:

Associate Professor Vineet R. Kamat, Chair
Assistant Professor SangHyun Lee
Assistant Professor Ann E. Jeffers
Assistant Professor David D. Wentzloff

To Momo and Fifi,
Who will always be missed.

To F.C. Internazionale Milano,
For inspiring me to never give up.

Acknowledgments

I am and will always be indebted to my advisor Dr. Vineet Kamat without whom none of my accomplishments would have been possible. His patience, encouragement and confidence in my abilities were invaluable in this journey. He has been a constant source of inspiration and the best advisor and mentor a student could have. The lessons I have learnt from him will remain with me and guide me for the rest of my life.

I would also like to acknowledge the contributions of my committee members Dr. SangHyun Lee, Dr. David Wentzloff, and Dr. Ann Jeffers. Their invaluable feedback has helped me shape the validation of various elements of my research. They have helped me redirect my research when appropriate, and guided it towards its ultimate direction of success.

I would like to thank Dr. John Everett for his most insightful lectures and thorough coverage of the basics of construction engineering and management practices. I would like to sincerely thank Dr. Sugih Jamin under whose guidance, my knowledge of computer graphics fundamentals increased tremendously. I am grateful to Mr. Andrew Morgan for building up my fundamentals in programming at a crucial juncture in my PhD journey. I would also like to thank all members of the OpenSceneGraph community for their support and proving why the Open Source movement is such a success.

I want to express my gratitude to Dr. Kamel Saidi and Ms. Geraldine Cheok for supporting me during my time at the National Institute of Standards and Technology, Gaithersburg, MD. I would like to thank Dr. Vinay Topkar, Mr. P.S. Tulpule, and Mr. V.G. Katre, who were instrumental in me beginning my graduate studies.

I want to thank my colleagues Manu Akula, Suyang Dong, Chen Feng, Nicholas Fredricks, Ali Golabchi, and Srinath Sridhar. Our discussions have been enriching, thought provoking, and entertaining. I wish you all the very best of success in your endeavors.

Last, but not least, I would like to acknowledge the support of my family and friends without which much of this would have been unattainable. My parents have been most caring, understanding and encouraging. It is impossible to imagine doing any of this without their unwavering love and support. Finally, I would like to thank my fiancée, Sri, for always being there, no matter what.

Table of Contents

List of Tables

List of Figures

List of Appendices

Abstract

An excavator unintentionally hits a buried utility every 60 seconds in the United States, causing several fatalities and injuries, and billions of dollars in damage each year. Most of these accidents occur either because excavator operators do not know where utilities are buried, or because they cannot perceive where the utilities are relative to the digging excavator. In particular, an operator has no practical means of knowing the distance of an excavator's digging implement (e.g. bucket) to the nearest buried obstructions until they are visually exposed, which means that the first estimate of proximity an operator receives is often after the digging implement has already struck the buried utility.

The objective of this dissertation was to remedy this situation and explore new proximity monitoring methods for improving the spatial awareness and decision-making capabilities of excavator operators. The research pursued fundamental knowledge in equipment articulation monitoring, and geometric proximity interpretation, and their integration for improving spatial awareness and operator knowledge. A comprehensive computational framework was developed to monitor construction activities in real-time in a concurrent 3D virtual world. As an excavator works, a geometric representation of the real ongoing process is recreated in the virtual environment using 3D models of the excavator, buried utilities and jobsite terrain. Data from sensors installed on the excavator is used to update the position and orientation of the corresponding equipment in the virtual world. Finally, geometric proximity monitoring and collision detection

computations are performed between the equipment end-effector and co-located buried utility models to provide distance and impending collision information to the operator, thereby realizing real time knowledge-based excavator operation and control.

The outcome of this research has the potential to transform excavator operation from a primarily skill-based activity to a knowledge-based practice, leading to significant increases in construction productivity and safety. This is turn is expected to help realize tangible cost savings and reduction of potential hazards to citizens, improvement in competitiveness of U.S. industry, and reduction in life cycle costs of underground infrastructure.

Chapter 1

Introduction

1.1 Introduction

Civil Engineering projects are unique in nature as they lead to the creation of an unstructured, dynamic, and continuously evolving work space (Son et al. 2008). While the US construction industry accounts for only around 7% of the total workforce, it is responsible for nearly 20% of all industrial fatalities (MacCollum, 1995). The most common accidents on construction jobsites are workers being struck by equipment or objects, workers falling into trenches or openings, electrocution, burns, and incidents involving the collapse of temporary structures. The nature of these accidents reveals that their cause can be attributed to constrained work spaces, lack of clear visibility, and lack of spatial awareness (Chi and Caldas 2012, Ruff 2004). Projects in crowded urban areas present project participants with narrow work spaces and a limited visibility of resources , both of which increase the possibility of collisions between equipment, workers, materials, and jobsite infrastructure (Teizer et al. 2010a, Cheng and Teizer 2011). Thus, generally speaking, workers and equipment operators have to function in less than optimal visibility conditions on a construction jobsite.

Some operations—such as excavation, trenching, and drilling—pose an inherent risk to equipment operators due to potential unintentional strikes with concealed or buried infrastructure. Heavy engineering and mining projects that involve the use of large

equipment increase the risk of collision as a result of blind spots on equipment and narrow haul roads (Teizer et al. 2010b, OSHA 2010). Some operations require equipment to be maneuvered remotely through a process called tele-operation in which operators control equipment with the aid of video cameras and wireless technology. Such operations introduce an additional obstacle in achieving spatial awareness due to the limited field of view afforded by on-board cameras (Chen et al. 2007).

As is evident from this discussion, certain operations add an additional layer of risk for equipment operators by diminishing the operators' ability to clearly perceive and analyze their working environment and its interactions. For example, excavator operators performing operations in the presence of underground utilities have the ever-present risk of striking buried utilities. Operators must rely on judgment and experience to avoid striking subsurface utilities covered by earth, dirt, and soil in the absence of equipment and infrastructure tracking. Another example of the lack of clear visibility is the case of drilling equipment operators carrying out operations on reinforced concrete slabs. As the underlying reinforcement steel and utility conduits are hidden from the operators' view, they cannot be certain of correct drilling locations in the absence of drill-bit position-orientation tracking with respect to obstruction locations.

The overall goal of this research was to improve the quality of information support provided to equipment operators in scenarios where the jobsite visibility and spatial awareness are reduced due to the type of task. The emphasis in particular is on aiding excavation operators in avoiding unintentional collisions with out-of-sight buried

infrastructure. This is achieved by designing methods to enable the representation of a real-world jobsite in a 3D virtual world that includes all entities that are critical to the operation. In the case of an excavation operation, such entities comprise the buried utilities and the moving excavation equipment that adequately define a 3D virtual scene to fully simulate the involved operation.

A series of interconnected computational frameworks has been developed to evaluate the designed methods and to enable real-time monitoring of excavation operations. The developed computational infrastructure encompasses the areas of real-time scalable 3D visualization, georeferenced 3D modeling of buried utilities, characterization and improvement of the overall life-cycle aspects of buried utility data, real-time proximity monitoring of entities present in the virtual world, creation of kinematically equivalent equipment models for use in 3D simulation, and transmission of sensor data from real to virtual worlds.

The computational frameworks are specifically designed so that they can be used independently to solve future research problems in construction operations monitoring and related areas. In addition, the concepts developed in each framework are implemented in corresponding software toolkits. Through these toolkits, the concepts developed can be demonstrated through real-world experiments. The individual toolkits representing their respective computational framework are combined to create a 3D visualization environment called SeePlusPlus. SeePlusPlus allows users to create an emulation of a real-world jobsite, and to provide operators with real-time visual guidance

and spatial awareness through proximity warnings. SeePlusPlus is the tangible product that manifests all of the technical contributions of this dissertation, and can be used to assist excavation operators through real-time visualization.

1.2 Background and Importance of the Research

As cities have grown larger and denser, so has their underground infrastructure. This underground infrastructure supplies water, electricity, gas, and telecommunication data. It also transmits sewer and storm water from their sources to treatment plants. The uninterrupted functioning of various utility networks serving a city is the only reason a city and its inhabitants can carry on their daily activities without any interruption to life and commerce. However, with aging buried infrastructure, pipe bursts, costly repairs, and lengthy supply outages have become commonplace (Belson and DePalma 2007). A large number of underground water supply pipes were built more than 50 years ago, and water mains in some urban areas have been in the ground for a century or longer (AWWA 2012). A similar problem has been reported in natural gas utility networks leading to explosions of gas pipelines (Homeland Security News Wire 2011).

There are an estimated 20 million miles (32 million km) of buried utilities in the United States (Anspach 2010). This is approximately 80 times the distance from the earth to the moon. There are more than one million miles (1.6 million km) of pipes in the United States that need to be replaced, which will cost over $1 trillion in the next 25 years (AWWA 2012). This problem is not limited to the U.S. alone; it is seen in all densely populated regions. For example, Hong Kong's fresh and saline water supplies are

provided through a network of approximately 4500 miles (7,200 km) of water mains, most of which are underground. Approximately 45 percent of the water mains were laid 30 years ago and are now approaching the end of their service life, becoming increasingly difficult and costly to maintain (Trenchless Australasia 2005). There is thus a global need to replace aging underground infrastructure, and this need in turn leads to a greater number of excavation-related operations in the presence of existing buried utilities.

It is therefore critical to investigate the risks involved in excavation operations carried out in close proximity to buried utilities. Excavation has been observed as being one of the leading causes of accidents involving buried utilities. The past decade (2002-2011) saw more than 800 reported incidents causing nearly $200 million in property damage in the U.S. (PHMSA 2012). The American Gas Association also considers digging near existing pipelines or conduits to be a leading cause of damage to pipelines (American Gas Association 2010).

It is common to find utilities running alongside each other with little or no clearance between them (Vickbridge 2007). As a result, any excavator striking a vicinal utility can lead to an explosion, electrocution, water leakage, and/or loss of telecommunications. Thus it is clear that excavator operators require additional spatial awareness of their surroundings, in this case the location of buried utilities in the vicinity of their equipment’s end-effector. Such spatial awareness can be provided conceptually through a combination of real-time visual assistance and proximity-based warnings.

Visual assistance can be provided through 3D virtual visualization. Such visualization, an abstraction of the real world, can contain the excavator, the buried utilities, and any other entities that would interact with the excavator during the course of operations. The advantage of virtual visualization is its ability to enable a viewer to perceive the world in ways that would not be possible with a conventional video camera. For example, in a virtual world, the opaque terrain can be made translucent and thus expose the locations of utilities lying beneath it. Similarly, perspective views of the jobsite can be made possible that would otherwise be impossible to achieve through conventional visualization or video monitoring.

Furthermore, through a virtual visualization, operators can have a better perspective of depth and horizontal distances. The occurrence of stacked utilities in the field gives 3D visualization an advantage over corresponding 2D displays. Thus the importance of visual assistance to excavator operators for improving safety and preventing unintended strikes during operations is clearly evident. In addition to visual guidance, operators can require additional real-time spatial information, such as the distance of their equipment's end-effector to unexposed utilities, the safe distance that may be traversed by the end-effector, etc. Spatial awareness can be provided to the operators through real-time proximity monitoring. The analysis can use 3D models representing the interacting entities on the jobsite, and can combine position and orientation data from the jobsite to compute distances between the entities. Through these computations, an operator can be warned as soon as the end-effector crosses a pre-determined safety threshold. The

analysis can also be used to provide warnings to the operator in the form of audio and visual interferences.

Other important aspects to consider for safe excavation are identification and representation of uncertainties. There are tremendous uncertainties associated with buried utility location data, and a utility is often not present in the real-world location that is indicated by as-built drawings (Sterling et al. 2009). The use of such uncertain data for visualization and proximity analysis can give a sense of false confidence to the operator, and can potentially mislead him/her into an unintentional strike. Thus it is essential for any location uncertainty associated with buried utilities to be represented in their visualization, and that this uncertainty be accounted for in proximity analysis.

Thus operators have a critical need to improve their spatial awareness of the overall jobsite, and they also need to receive timely warnings to prevent unintentional utility strikes. There is an additional urgent need to represent uncertainty information in an easy-to-understand manner and include it in all analyses. In practical terms, what excavator operators require is a tool that will assist them in carrying out their task in a safer manner. This dissertation addresses the aforementioned theoretical and practical needs, and pursues research aimed at providing contextual information assistance to equipment operators through a real-time 3D visualization and jobsite emulation framework.

The research also develops a proximity-monitoring framework that provides timely warnings to operators so they may prevent potential strikes. Finally, the location

uncertainty associated with buried utility data is represented in both the visualization framework as well as the proximity analyses, thus preventing scenarios of false confidence.

1.3 Real-time 3D simulation of construction operations

The dynamic, unstructured nature of a construction jobsite, together with the presence of workers and equipment often alongside each other, makes a jobsite a high-risk environment. A representation of construction jobsites in a 3D virtual setting is said to be successful when the virtual representation adequately resembles the way in which processes are carried out in the real world (Kamat and Martinez 2001). Visualization can occur in both two and three dimensions. An example of a 2D visualization system is Proof, which can be used to visually represent simulations ranging from mining and construction, to material handling and airport traffic (Wolverine Software 2012). However, 2D representations lack the realism that is present in 3D visualization (Kamat and Martinez 2001). Human beings depend on sight as one of their key senses, and in general can process graphic information more quickly than other forms of information (Rohrer 2000). Furthermore, it is felt that only a 3D visualization can communicate the spatial orientation of equipment and its end-effectors in relation to a construction jobsite.

The use of 3D visualization for representing discrete event simulations in a graphical environment has been successfully demonstrated through the VITASCOPE 3D visualization and animation language (Kamat and Martinez 2008). 3D visualization has also been implemented for virtually viewing a facility being constructed over its entire

construction phase by animating the transformation of the built environment over the entire period. This is called 4D CAD, and is accomplished by linking the 3D CAD model representing building/facility design to its associated construction schedule (McKinney et al. 1996). All of the above are examples where visualization was post-processed (i.e., they did not represent concurrently ongoing operations in the real world). This research introduces the concept of real-time 3D visualization for monitoring construction jobsites as operations are carried out.

3D visualization has tremendous scope for monitoring ongoing construction operations in real-time by providing assistance to equipment operators and site managers. Thus the visualization is an abstracted representation of the real world in a 3D virtual world. This virtual world gets updated in real-time through sensor-based feedback from the real world, and provides useful information to users such as impending collisions and adherence to required safety thresholds. As the representation is virtual, a different perspective of the jobsite can be provided to users, which is not possible in the real world. Examples of these different perspectives include the ability to view: (1) buried utilities, (2) infrastructure concealed in walls and ceilings, or (3) perpendicular traffic at narrow haul road intersections—all by changing the transparency of the occluding entity in the virtual world. Hence, through such visualization, assistance for avoiding collisions with workers, other equipment, infrastructure, and/or materials can be provided and can supplement the accident prevention technologies required by current safety regulations (OSHA 2010).

Conventionally, visual guidance to operators would be provided in the form of a video stream from a camera mounted on a piece of equipment, such as a backup camera or side-facing camera. While video cameras help to provide additional viewpoints (backup camera, e.g.) to reduce the effect of operators' blind spots, they do not provide a cohesive view of the equipment's surroundings. This cohesive view *is* afforded elsewise, though—by a virtual over-the-shoulder or top-view camera in a 3D world. The views from a camera mounted on a piece of equipment suffer from a narrow field of view (Huber et al. 2009) and disjointed viewpoints. A virtual scene, on the other hand, allows a camera to be placed at any location in the scene, and provides a user-defined view. Figure 1.1 shows a comparison of the views from a conventional video camera and a virtual camera for surrounding views from an excavator.

Fig. 1.1: Comparison of conventional view from a backup video camera view (left: Source: http://heg.baumpub.com/products/8989/backup-camera-designed-for-heavy-equipment) and that from a 3D virtual view

Unlike 3D visualization, conventional/camera-based visualization has its limitations as it can only provide visual assistance. This means that it is not possible to easily carry out proximity queries between objects seen in the video stream. Furthermore, for tasks such as drilling, excavating, and trenching, the view from an additional video camera cannot provide a significantly superior view of the surroundings compared to that seen directly by the operator. In such cases, a 3D virtual environment makes it possible to alter the transparency of the occluding layer and thus expose the underlying infrastructure such as reinforcement bars, pipes, conduits, and/or ducts. In addition, 3D visualization enables users to set up distance and collision queries between two or more entities that may be present in a scene, thus providing distance information.

1.4 Research Hypothesis and Scope

The research hypothesis is: If real-time 3D visualization and geometric proximity monitoring provide improved visibility guidance and spatial awareness, then a framework simulating a real-world excavation operation as a 3D virtual world through sensor-based equipment tracking, georeferenced 3D modeling of buried utilities, and real-time proximity queries between virtual 3D entities can help prevent unintended collisions between equipment end-effectors and buried utilities.

The emphasis in this research is on creating a 3D virtual world that is representative of the real-world jobsite consisting of 3D equipment models, georeferenced 3D models of buried utilities, and any other static and dynamic entities that are involved in the

operation. The 3D virtual world utilizes position and orientation data from the jobsite, and updates dynamic entities corresponding to the real-world sensors. Finally, the framework provides real-time proximity analysis through the use of geometric 3D models and updates on position and orientation. The research also includes the characterization of uncertainties present in the data in both the visualization and proximity analysis aspects. The research also evaluates computations made in the virtual world with corresponding real-world values that represent the ground truth equivalent.

1.5 Research Objectives

The objectives of this research were as follows:

- Investigate the current technologies in place to prevent collisions between underground infrastructure and excavator end-effectors
- Analyze the limitations in current accident prevention practices for buried utility excavation operations
- Understand the life-cycle that buried utility data passes through—characterizing the improvement areas and factors for consideration—while creating 3D models of buried utilities
- Create georeferenced 3D models of buried utilities that also capture location uncertainty associated with data through an easy-to-understand visual representation
- Develop a framework to represent a real-world operation in a 3D virtual world through sensor-updated dynamic entities and georeferenced static entities, while maintaining an adequate level of abstraction for effective representation

- Evaluate various visualization methodologies and implementations for their suitability in creating a 3D virtual world simulating real-world construction operations and equipment in real time
- Understand the kinematic articulation of construction equipment and issues surrounding their dynamic representation in a virtual environment
- Enable an interface between real-world sensors and 3D equipment articulations to allow position and orientation updates of equipment components through sensor streaming data
- Create a generic user interface to allocate sensor streams to a wide array of 3D articulated equipment to enable real-time visualization of an arbitrary construction operation
- Review proximity query and monitoring methodologies in computer graphics literature to identify the most suitable approach for providing computational support for a 3D visualization framework
- Create a real-time geometric proximity monitoring framework to analyze the interactions between entities involved in the real-time simulated operation, and to provide distance and collision information
- Develop a scalable framework that can be used to simulate any real-world construction operation involving a combination of static and dynamic entities with participants requiring real-time visual assistance and proximity feedback

The end result of pursuing these research objectives is a 3D visualization toolkit known as SeePlusPlus. The name derives inspiration from the combination of the toolkit's ability

to allow a user to virtually see infrastructure and objects that would otherwise remain occluded in the real world, and the use of the C++ programming language for its implementation. SeePlusPlus is designed as a combination of multiple frameworks and modules that provide the functionality required to create a 3D emulation of real-world operations and provide audio and visual warnings to users when user-defined safety thresholds are breached.

1.6 Research Methodology

The methodology was composed of different research thrusts, each having a specific focus. The breakdown of the overall research methodology into its key components is shown in Figure 1.2.

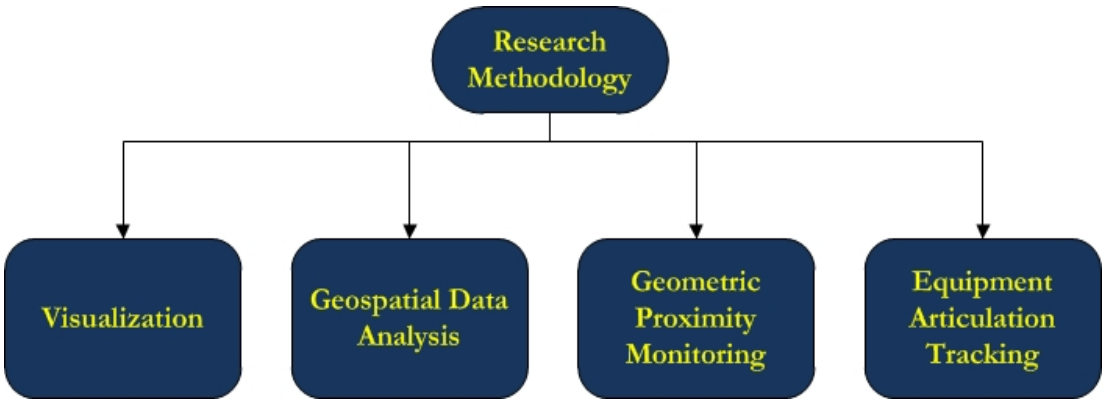

Fig. 1.2: Primary Thrusts in Research Methodology

Visualization:

- A framework was designed to represent a real-world operation in an entirely virtual 3D environment by integrating dynamic 3D models, real-world position and orientation data, and georeferenced 3D (static) models

- The framework was designed as a real-time visualization to assist equipment operators and site managers in their monitoring of activities on the jobsite
- Even though the target application for this framework was mechanized excavation operations, emphasis was given to ensure that the end result was scalable and open-ended, such that the framework would be applicable to any generic construction operation involving a combination of static and dynamic entities

Geospatial Data Analysis:

- The current mandatory one-call marking system in which any excavation crew is required to have the proposed digging area pre-marked prior to excavation was reviewed, and limitations with the current process were identified
- The lack of permanent visual guidance, the absence of depth values and spatial awareness, and the utility-specific location uncertainty information were identified as the key limiting factors to be targeted by the proposed framework
- Reviewed the current practices and limitations for the entire life-cycle that buried utility data passes through, and classified the life-cycle into Collection, Archival, Update and Visualization stages
- Created a list of best-practices for the effective visualization of buried utilities, which is presented as the IDEAL framework due to its requirement of **I**nteractivity, **I**nformation Richness, **D**imensionality, **A**ccuracy, and Extensibi**l**ity in any visualization implementation for viewing buried utilities and related operations

- Designed a 3D modeling toolkit to produce georeferenced, color-coded (APWA 2005), attribute-rich and location-uncertainty-inclusive models of buried utilities using XML, GIS, or CAD data for input

Geometric Proximity Monitoring:

- Reviewed computer graphics literature in relation to proximity queries and associated concepts such as bounding volumes, tree traversal, and overlap tests
- Identified the primary requirements and most suitable option for minimizing time and memory footprint for implementation in a mobile, low computing power unit
- Developed a framework to include 3D geometry models, position, and orientation updates to perform **pro**ximity analysis, **to**lerance checks, and **col**lision detection between a pair of entities. The framework is accordingly named PROTOCOL
- Implemented PQP library (PQP Gamma 1999) for geometric proximity analysis within the PROTOCOL framework
- Designed and implemented a user interface to represent the functionalities present in the PROTOCOL framework for creating one-one and one-many proximity queries (Euclidean distance, collision check, and/or tolerance zones) between any two entities present in the virtual scene at run-time and a-priori

Equipment Articulation Tracking:

- Studied the forward kinematics principles governing articulated equipment (e.g., excavators) to apply them in creating kinematically equivalent 3D models

- Focused on creating a generic method applicable to multiple equipment configurations and types
- Designed and implemented a tool for creating kinematically equivalent 3D models of articulated equipment such that their dimensions, rotation, and position of end-effector match that of their real-world counterparts
- Developed a framework for connecting sensors from the real-world jobsite to update corresponding components of articulated 3D models in the virtual world through allocation of sensor streams
    - Designed the capability to allocate sensor streams to generic articulated equipment (e.g., excavators, cranes, etc.) without being restricted to working with a single type of equipment
    - Implemented framework using socket-based server-client relationships

1.7 Dissertation Outline

This dissertation is a compilation of scientific manuscripts that document the research involved in designing SeePlusPlus and its supporting modules. In particular, chapters 2 through 5 are stand-alone papers that describe details of individual scientific questions answered, challenges encountered, alternatives considered, and methods adopted in addressing a specific research issue or a set of related issues.

All chapters have been written such that they can be read and clearly understood by a technically literate audience from diverse fields, including those without any prior exposure to 3D computer graphics and/or computer programming. In addition to the

manuscripts presented in subsequent chapters, the preliminary work leading to the development of the final framework is described in Talmaki et al. (2010a, 2010b). Each manuscript focuses on the technical details of a single module of the overall framework, as well as any experiments conducted to validate it.

- **Chapter 2 - Real-Time Hybrid Virtuality for Prevention of Excavation-Related Utility Strikes**

This chapter provides an overview of the problem and the proposed methodology of using real-time 3D visualization to simulate a real-world operation. The key questions it answers are as follows: 1) What methods are currently used to prevent accidents during excavation operations involving buried utilities? 2) How can 3D visualization be used to represent a real-world jobsite? 3) What elements are needed in a 3D visualization in order to create a valid representation? 4) What associated research problems need to be addressed in order to achieve the objective of assisting excavation equipment operators?

First, the procedures and regulations in place to provide excavation crews with buried utility location information were studied, and their limitations analyzed. Second, a concept of Hybrid Virtuality was introduced that enables a 100% virtual environment to be an abstract representation of the real world by allowing the scene modeler to interactively pick elements that are crucial for the operation. Third, the requirements for creating a virtual world representing an excavation jobsite were systematically studied. Fourth, additional research thrusts beyond the core 3D visualization were identified.

- **Chapter 3 - Geometric Modeling of Geospatial Data for Visualization-Assisted Excavation**

This chapter focuses on the buried utility aspect of the proposed framework. In creating a 3D virtual world, the buried utilities form one half of the relationship that needs to be monitored in real-time, with the other half being the equipment end-effector. The main questions addressed in this chapter are as follows: 1) What are the reasons for the uncertainty associated with buried utility data? 2) What are the different life-cycle stages that buried utility data passes through, and what are some inherent limitations in the current practice? 3) How can buried utility data be effectively visualized for both monitoring and maintenance purposes? 4) How can buried utility data be used in the proposed 3D visualization framework for proximity monitoring purposes?

First, a review of the current practices concerning buried utility data was carried out. Second, the entire life-cycle that buried utility data passes through was investigated and divided into the distinct phases of Collection, Archival, Updating, and Visualization. The limitations and recent improvements in each of the phases were also documented. Third, a framework for the effective visualization of buried utility data was created so that any visualization prescribing to the requirements can provide maintenance personnel and equipment operators with an improved visualization of buried infrastructure. Fourth, the need for 3D models of buried utilities in 3D simulations of excavation operations was reviewed and implemented through the creation of georeferenced 3D utility models from GIS and XML data sources.

- **Chapter 4 - Active Geometric Proximity Monitoring of Visibility-Constrained Construction Processes Using Real-Time Graphical Simulation**

The thrust of this chapter is the proximity-monitoring framework that uses 3D models present in the virtual world together with their real-world position and orientation to compute Euclidean distance, and any collisions between them. The primary questions addressed in this chapter are: 1) What are the technological approaches possible for monitoring the position and orientation of articulated equipment (e.g., excavators) on construction jobsites? 2) How can the 3D models used in a real-time 3D visualization framework be used for geometric proximity queries? 3) What are the available algorithms to optimize speed, time, and computational demands for creating a proximity-monitoring framework?

First, a thorough review of previous methodologies for tracking equipment on a construction jobsite was carried out. The limitations and advantages of these methodologies were identified. The combinatorial use of position and orientation sensors to comprehensively represent equipment's real-world position and articulation orientation was investigated. Second, fundamental principles of computer graphics, scene graphs, and kinematics were studied to use real-world sensor data to update the position, roll, pitch, and yaw of equipment components in a 3D virtual world. Third, efficient methods for computing geometric interference and Euclidean distance between 3D models were examined and incorporated into a framework for real-time proximity monitoring between any combination of static and dynamic entities in a 3D scene. A validation experiment using indoor Global Positioning System carried out at the National Institute of Standards

and Technology is also presented to characterize the positional errors and latency introduced by the framework.

- **Chapter 5 - Evaluation of Sensor Retrofits for Real-Time Graphical Emulation of Articulated Construction Equipment**

This chapter focuses on the dynamic half of the tracked relationship from the real world. The challenges involved in creating a seamless and accurate link between the real and virtual worlds are examined. A framework for using real-world position and orientation sensor data is developed and is presented in this chapter. This chapter seeks to answer the following research questions: 1) How is equipment monitored on construction and mining jobsites? 2) What are the key requirements of a framework enabling the use of real-world position and orientation data in a virtual world? 3) What are some of the challenges involved in simulating dynamic equipment accurately in a 3D virtual world?

First, a review of existing equipment monitoring methodologies was performed. Second, the need for a framework capable of handling any configuration of equipment, and position and orientation sensors was clearly identified by recognizing the wide array of articulated equipment types, and position and orientation sensors available for tracking. Third, the key requirements for accurately representing a piece of equipment and updating its 3D model in a virtual world (e.g., kinematic and dimensional equivalence) were examined and incorporated into the generic scalable framework that was developed. In addition, a validation experiment was presented to demonstrate the use of real-world

orientation sensor data inside of a 3D virtual world for simulating the motion of a backhoe's articulated arm in real time.

- **Chapter 6 - Conclusion**

This chapter summarizes the contributions and achievements of this research. The exercise conducted to validate the work's tangible product (i.e., SeePlusPlus and its associated components) is presented in detail. The chapter also provides other usage scenarios by presenting them as virtual scenarios. The chapter concludes the dissertation by suggesting specific directions for future research.

1.8 References

American Gas Association (AGA) (2010). “What causes natural gas pipeline accidents?”, <http://www.aga.org/KC/ABOUTNATURALGAS/CONSUMERINFO/Pages/CausesofNGPipelineAccidents.aspx> (08/21/2012)

American Water Works Association (AWWA) 2012. “Buried No Longer: Confronting America’s Water Infrastructure Challenge”, <http://www.awwa.org/files/GovtPublicAffairs/GADocuments/BuriedNoLongerCompleteFinal.pdf > (08/21/2012)

American Public Works Association (APWA) (2005). “Uniform Color Code”, <http://www2.apwa.net/documents/About/TechSvcs/One-Call/COLORCC.PDF> (11/19/2011)

Anspach, J. (2010). “Collecting and Converting 2-D Utility Mapping to 3-D”, Pipelines 2010. September 2010, 278–284.

Belson, K., DePalma, A (2007). “Asbestos and Aging Pipes Remain Buried Hazards”, NYTimes, July 19, 2007, <http://www.nytimes.com/2007/07/19/nyregion/19steam.html> (08/21/2012)

Chen, J. Y.C., Haas, E. C., and Barnes, M. J. (2007). “Human performance issues and user interface design for teleoperated robots”, IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, 37: 1231–1245.

Cheng, T., and Teizer, J. (2011). “Crane Operator Visibility of Ground Operations”, 2011 Proceedings of the 28th ISARC, Seoul, Korea, Pages 699–705.

Chi, S. and Caldas, C. (2012). “Image-Based Safety Assessment: Automated Spatial Safety Risk Identification of Earthmoving and Surface Mining Activities”, Journal of Construction Engineering and Management, 138(3), 341–351.

Homeland Security News Wire (2011). “Aging U.S. natural gas pipelines are ticking time bombs”, <http://www.homelandsecuritynewswire.com/aging-us-natural-gas-pipelines-are-ticking-time-bombs> (08/21/2012)

Huber, D., Herman, H., Kelly, A., Rander, P., and Warner, R. (2009). “Real-time Photorealistic Visualization of 3D Environments for Enhanced Teleoperation of Vehicles”, Proceedings of the 2nd International Conference on 3D Digital Imaging and Modeling, Kyoto, Japan.

Kamat, V. R., and Martinez, J. C. (2008). “Software Mechanisms for Extensible and Scalable 3D Visualization of Construction Operations”, Advances in Engineering Software, 39 (8), Elsevier Science, New York, NY, 659–675.

Kamat, V. R., and Martinez, J. C. (2001). “Visualizing simulated construction operations in 3D.” Journal of Computing in Civil Engineering, Vol. 15, No. 4, 329–337.

MacCollum, D.V. (1995). “Construction Safety Planning”, John Wiley & Sons, New York, pp. 53–54.

McKinney, K., Kim, J., Fischer, M., and Howard, C. (1996). ‘‘Interactive 4D-CAD’’ Proc., 3rd Congr. on Comp. in Civ. Engrg., ASCE, New York, 383–389.

Occupational Safety and Health Administration (2010) <http://www.osha.gov/pls/oshaweb/owadisp.show_document?p_table=INTERPRETATIONS&p_id=27308 > (06/17/2012)

PHMSA (2012), “Significant Pipeline Incidents through 2010 By Cause” <http://primis.phmsa.dot.gov/comm/reports/safety/PSI.html> (08/21/2012)

PQP Gamma (1999) “PQP – A Proximity Query Package” <http://gamma.cs.unc.edu/SSV/> (08/22/2012)

Rohrer, M.W. (2000). “Seeing is believing: the importance of visualization in manufacturing simulation”, Proceedings of the 32nd conference on Winter simulation, December 10–13, 2000, Orlando, Florida.

Ruff, T. M. (2004). “Evaluation of devices to prevent construction equipment backing incidents”, SAE Commercial Vehicle Engineering Congress and Exhibition, Chicago, IL, 2004-01-2725, 10.

Son, H., Kim, Ch., Kim, H., Choi, K.-N., 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 (2008), pp. 232–237.

Sterling, R.L., Anspach, J., Allouche, E., Simicevic, J., Rogers, C.D.F., Weston, K., and Hayes, K. (2009). Encouraging Innovation in Locating and Characterizing Underground Utilities, SHRP 2 Report S2-R01-RW, Transportation Research Board, Washington, D.C.

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

Talmaki, S., Dong, S., and Kamat, V.R. (2010). "Geospatial Databases and Augmented Reality Visualization for Improving Safety in Urban Excavation Operations",

Proceedings of the 2010 Construction Research Congress: Innovation for Reshaping Construction Practice, American Society of Civil Engineers, Reston, VA, 91–101.

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

Teizer, J., Allread, B.S., Fullerton, C.E., Hinze, J., (2010b). “Autonomous pro-active realtime construction worker and equipment operator proximity safety alert system”, Automation in Construction 19 (5), 630–640.

Trenchless Australasia (2005), “Trenchless Technology in Hong Kong”, Trenchless Australasia, March 2005 Issue, <http://trenchless-australasia.com/news/trenchless_technology_in_hong_kong/002008/> (08/21/2012)

Vickbridge, I. (2007), “The Trenchless challenge in urban Hong Kong”, Trenchless Australasia, September 2007, <http://trenchless-australasia.com/news/the_trenchless_challenge_in_urban_hong_kong/001643/> (08/21/2012)

Wolverine Softwre (2012), Proof Animation Software, <http://www.wolverinesoftware.com/ProofOverview.htm> (08/22/2012)

Chapter 2

Real-Time Hybrid Virtuality for Prevention of Excavation-Related Utility Strikes

2.1 Introduction

The problem with underground utility lines being struck by excavating equipment is long-standing and significant. This is underlined by the congressional Transportation Equity Act for the 21st Century, TEA 21, Title VII, Subtitle C, SEC. 87301, which states that, *"…unintentional damage to underground facilities during excavation is a significant cause of disruptions in telecommunications, water supply, electric power, and other vital public services, such as hospital and air traffic control operations, and is a leading cause of natural gas and hazardous liquid pipeline accidents."* Citizens often take for granted the complex utility networks that lie underground in all major industrial and urban areas. Beneath the ground surface, there exists an extremely complicated web of cables, wires, pipes, and other conduits that transport water, natural gas, electricity, telecommunications, waste, and steam. Over the years, there have been numerous major accidents in which excavation crews have struck underground utility lines, leading to loss of life and/or damage to property.

In 1995, a construction crew driving piles for a new parking deck accidently crushed the high voltage underground electric cables serving Newark International Airport. As a result of the power loss, jet-fuel pumps, escalators, and automatic doors became inoperable. Hundreds of flights were cancelled and thousands of passengers were left

stranded until the cables could be fixed (New York Times 1995). In 2007, an excavation contractor's crew struck a high-pressure gas main in Cary, North Carolina, creating a 100-foot-high fireball that burned for nearly 6 hours. This led to an emergency evacuation of nearby residents and the closing of major roads (WRAL 2007). In June 2010, an excavation contractor hit an unmarked 14-inch gas pipeline near Darrouzett, Texas. Two workers were killed and one was seriously injured (Wilder 2010). It is estimated that there are nearly 500,000 utility strikes per year in the United States. An analysis of this number reveals that it equates to as much as one utility line being struck every minute (i.e. 365*24*60) in the United States (CGA 2008). The United States Pipeline and Hazardous Materials Safety Administration records show that the period from 2001 through 2010 saw a total of 2,770 serious incidents, of which nearly 20% were excavation-related incidents. From the 544 excavation-related accidents, 37 fatalities, 152 injuries, and close to $200 million in property damage resulted (PHMSA 2012a).

2.2 Current state of excavation related damage prevention

Every state in the United States has a one-call law that requires anybody carrying out an excavation operation to pre-mark the location of utility lines within the excavation zone. This law is implemented through "One-Call Centers" that serve as clearinghouses for utility pre-marking requests from excavation crews. The one-call centers contact their member companies consisting of the utility service providers lying within the center's jurisdiction. The utility lines' locations are marked on the surface using a combination of flags, stakes, and spray paint. This workflow is illustrated in Figure 2.1.

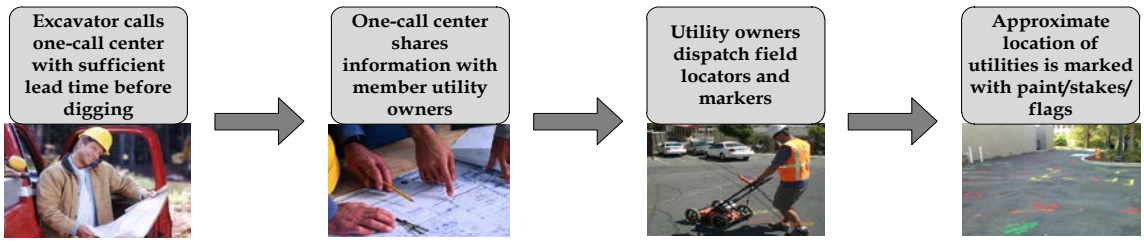

Fig. 2.1: Workflow description for a typical one-call center ticket

Even with the introduction of the one-call pre-marking system, accidents continue to happen. The year 2010 saw over $22 million in damage caused by excavation-related accidents (PHMSA 2012b). Thus additional accident prevention techniques need to be adopted to complement the existing methods. The authors have identified three limitations of the current damage prevention techniques. The first is the lack of depth information for the utilities. Markings provided by one-call agencies only identify the horizontal location of the underground utilities in plan view, but do not provide any depth value (Miss Utility Call Center Information 2012). The presence of other utilities, lack of accurate documentation, and effect of weather all create a high level of uncertainty. As a result, depth values of utility lines are not marked (Miss Utility Call Center Information, UTNE 2010). Consequently, excavation crews are unsure of the safe digging depth, often resulting in accidental utility strikes.

The second limitation concerns the degree of accuracy and quality level of data used for one-call utility markings. Accuracy relates to the precision of the utility location values, which depend upon data quality factors such as the skill level of personnel involved in the data collection process, the accuracy of the instrument used to collect data, and the uncertainties of existing datasets used in the representation of utility locations. The utility

location markings only indicate the approximate location of the lines. The pre-marked area is equal to the diameter of the utility line plus 18 inches on either side of its outer diameter. The utility line can lie anywhere within this zone. For many stakeholders, this is beyond their acceptable accuracy limit (Thomas et al 2007). Most of the problems plaguing current utility marking techniques stem from the accuracy of the maps and drawings used for marking sub-surface utilities. Many cities in the United States have underground utility lines that are over half a century old (UTNE 2010, Shevlin 2011). The discrepancy between as-designed or as-built data in the drawings, and the location of the utility lines in the real world, is a prime cause for wrongly marked utility lines. The accuracy of as-built drawings for the location and depth of such utility lines cannot be considered reliable. This causes the markings to be approximate and inaccurate in nature (Bernold 2005).

The third limitation concerns the temporary nature of the markings that are no longer visible once the top surface has been removed (Ron May, DTE Energy 2009, Personal Communication; Eric Urbain, MISS DIG Systems Inc. 2011, Personal Communication). When excavation is carried out in phases on large job sites, an operator may be required to resume operations after a time gap, with the previous utility markings now absent due to the top surface being excavated. This results in the excavator operator relying on judgment and memory rather than specific information. Markings such as flags and stakes are unreliable in rough weather, as their position may be unknowingly shifted or removed due to wind and rain. Carrying out excavation without having the locations re-marked increases the risk of a utility line strike.

2.3 Previous Studies

There has been significant research interest in alleviating the problems associated with buried utility location. There has also been federal interest in utility detection and excavation safety. The Federal Laboratory Consortium (FLC) report on utility locating technologies identified several limitations in current utility location techniques (Sterling 2000; Bernold 2005). The report states two overarching goals for future location techniques. The first goal is to avoid third party damage to existing underground utilities due to unknown or wrongly located utilities. The second goal is the development of location techniques to accurately identify all utilities regardless of their size, depth, and materials, and regardless of the presence of other utilities.

The authors classify buried utility location techniques into two sub-types, as depicted in Figure 2.2. The first group uses a combination of geophysical technologies in order to accurately determine the location and type of buried utilities. The first category is referred to as the multi-sensor approach. The second category uses a combination of geospatial databases, tracking technology, and computer graphics visualization to depict the position of the sub-surface utility lines with respect to the equipment operator and the excavation crew. Category two can thus be referred to as the Information Technology approach.

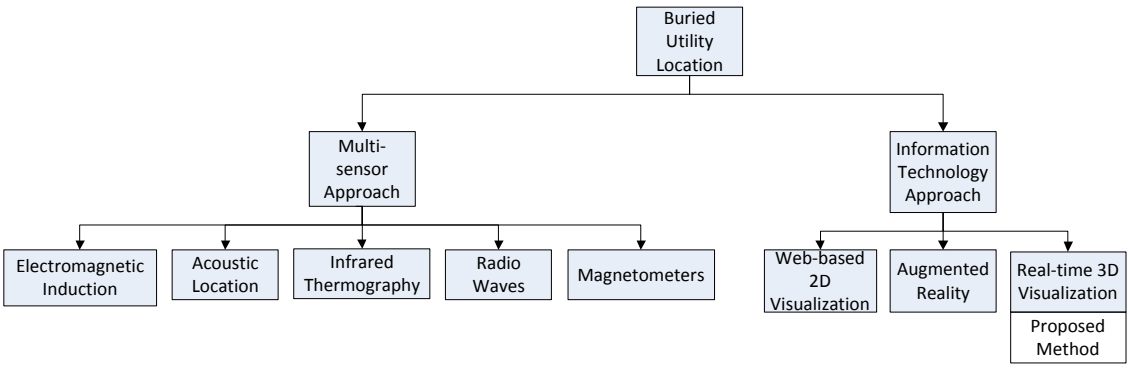

Fig. 2.2: Buried Utility Location and Visualization Techniques

The multi-sensor approach is based on the combined application of geophysical technologies such as electromagnetic induction, radio waves, magnetometers, infrared thermography, and acoustic location. It has been shown that no single geophysical technique can identify all utility types in varying soil conditions (Costello et al 2007, Bernold 2005). The combined output of multiple sensors can be used to provide excavation crews with accurate information regarding the position and depth of buried utilities present under an excavation site. A description of the geophysical techniques used for buried utility detection can be found in Talmaki et al. (2010).

The information technology approach is based on the use of geospatial databases to store buried utility data that is then displayed to the user in a 2D or 3D visualization. A two-dimensional display of utility data from heterogeneous data sources through a web-based service has been demonstrated and deployed successfully (Beck et al 2009). The rapid advancements in the computing power of mobile devices—such as smart phones, personal digital assistants (PDA), and tablet PCs—means that these devices can now run

computer graphics applications that were previously only possible on desktop workstations.

Another approach has used Augmented Reality Visualization. In this approach, virtual buried utility models are superimposed over the real-world view, thus giving the user a mixed view of the real world and virtual underground utility lines. This has been demonstrated to be a potentially useful tool for utility inspectors and maintenance crews (Evans et al. 2003, Junghanns 2008, Schall et al. 2009, Talmaki et al. 2010). The effectiveness of such applications is dependent on the accuracy of the geospatial utility database, as well as the accuracy of the tracking technology used. The realistic and intuitive output from such information technology applications can mislead the user if the source data is inaccurate. The user must be warned of inaccuracies in the data being viewed (Roberts et al 2006). At the same time, there is an increasing use of laser tracking and global positioning systems for mapping the position of buried utility lines (Patel and Chasey 2010). This leads to geospatial databases having accuracy within the desired limits, lending greater credibility to the various information technology approaches.

2.4 Proposed Framework

The presence of soil and pavement obscures underground utilities from an excavator operator's view. It is almost impossible for an excavator operator to be certain of the location of the utility lines in the vertical and horizontal planes without knowing their precise depth below the surface and their path on the surface, respectively. As this is one of the major drawbacks of current utility marking practices, there is a critical need for a

method that displays the precise location of the utility lines in plan and elevation without being affected by physical obstructions. It is also necessary to supply real-time proximity data and utility attribute information to assist operators in preventing accidental utility line strikes.

The authors propose a novel method of Hybrid Virtuality (HV) to address the problem described. The term Hybrid Reality has been used in the past to describe scenes having a combination of real-world elements and virtual objects. Such Mixed Reality (MR) scenes lie between being completely real and completely virtual, somewhere along the Virtuality Continuum (Miligram and Ballantyne 1997). The term "Virtual Reality" does not adequately describe a virtual scene emulating a real-world job site through the use of GIS and CAD data, position and orientation sensors, and user input through human interface devices. Hence the term Hybrid Virtuality has been used to describe a scene made up of 3D models that represent real-world elements that rely on tracking information from sensors instrumented on real-world entities (e.g., terrain, surface and underground infrastructure, and on-site equipment).

Hybrid virtual scenes utilize sensor data and user input from the real world to update the virtual objects in the scene. The aim of a Hybrid Virtual scene is to be a representation of the real-world job site at every instant, and to provide the user with additional information that may not be possible from a conventional real-world view. Virtual scenes such as these will from now on be referred to as hybrid virtual (HV) scenes. The authors investigate the feasibility and concepts involved in an HV scene to provide excavation

crews with an un-obstructed view of the underground utility lines, and with collision detection warnings to prevent accidental buried utility strikes.

2.5 Technical Approach

The HV simulation design is shown in Figure 2.3. It consists of four main components, namely: a Simulation and Graphics Engine, GIS and CAD data, Sensor Input, and User Input. The core of a visual simulation is the graphics engine. It is responsible for rendering the scene of the virtual jobsite. It also performs other related tasks associated with the scene such as displaying warnings when the proximity between any two entities—an excavator digging implement and underground utilities, for example—is within the safety threshold. It also provides the operator with vital information, such as attribute details of a utility line, that may not be evident in the real-world view.

GIS and CAD data are required to represent a typical jobsite. Underground utility lines data, temporary structures, semi-completed structures, and construction equipment are just some of the examples of the type of GIS and CAD data that is required to represent a jobsite in an HV simulation. However, the data needs to be accurate and current; otherwise, the simulation will not be an accurate emulation of the real-world job site. The sensor input depends on the type of equipment used and the real-world operation being simulated. Examples of sensors on a jobsite are GPS tracking sensors and on-board equipment sensors such as strain gauges and accelerometers. The output of these sensors is used to keep the simulation up-to-date with the real world, leading to a near-real-time scenario (Stone et al. 1999).

User input comes from human interface devices (HID) such as a keyboard, joystick, or mouse. User input is used to perform a number of tasks, such as the display of on-demand information like utility attributes, changing the transparency level of 3D models like the terrain, and altering the viewing position of the user within the HV simulation. In addition, HID can be used as a replacement for, or in combination with, sensor input to provide commands to update the position and orientation of equipment in the scene for simulation set up purposes. Thus HV simulations can be categorized as Human in-the-Loop simulations (DoD 1998). The rest of this chapter describes HV simulations in detail and how they can be used to alleviate the problem of underground utility lines being struck by excavator operators.

2.5.1 Framework Overview and Data Flow

The proposed method's data flow between its components is represented in Figure 2.3. The examples of each component reflect those required for an HV simulation representing an excavation jobsite. The input to the simulation and graphics engine is the GIS and CAD data, user input, and sensor input. The output consists of real-time visuals showing the virtual jobsite, attribute details of GIS and CAD models, and proximity values between the various entities present in the HV simulation. As the sensor and user input components update the GIS and CAD data in the graphics engine in real-time, the HV simulation serves warnings if two entities come within the safety tolerance distance of each other. The following sections describe the details of each component of an HV simulation and the specifics involved for an excavation jobsite.

2.5.2 Sensor input

Research at the National Construction Automation Testbed (NCAT) has demonstrated the usefulness of 3D simulations using input from on-board equipment sensors. Such visual simulations can provide far more information to a site manager or equipment operator than a video of the scene from a camera (Stone et al. 1999). The primary use of sensor input is to update the position and orientation of the 3D CAD objects in the HV scene. Thus the choice of sensors depends upon the equipment being used. For example, the translation motion of a dump truck can be recorded using GPS sensors placed on it. Input from sensors has another valuable use for accident investigations. The data stream from on-board equipment sensors can be logged and re-used for replaying the events that lead to an accident using the visual environment of the HV simulation. Such replaying can be referred to as a trace simulation.

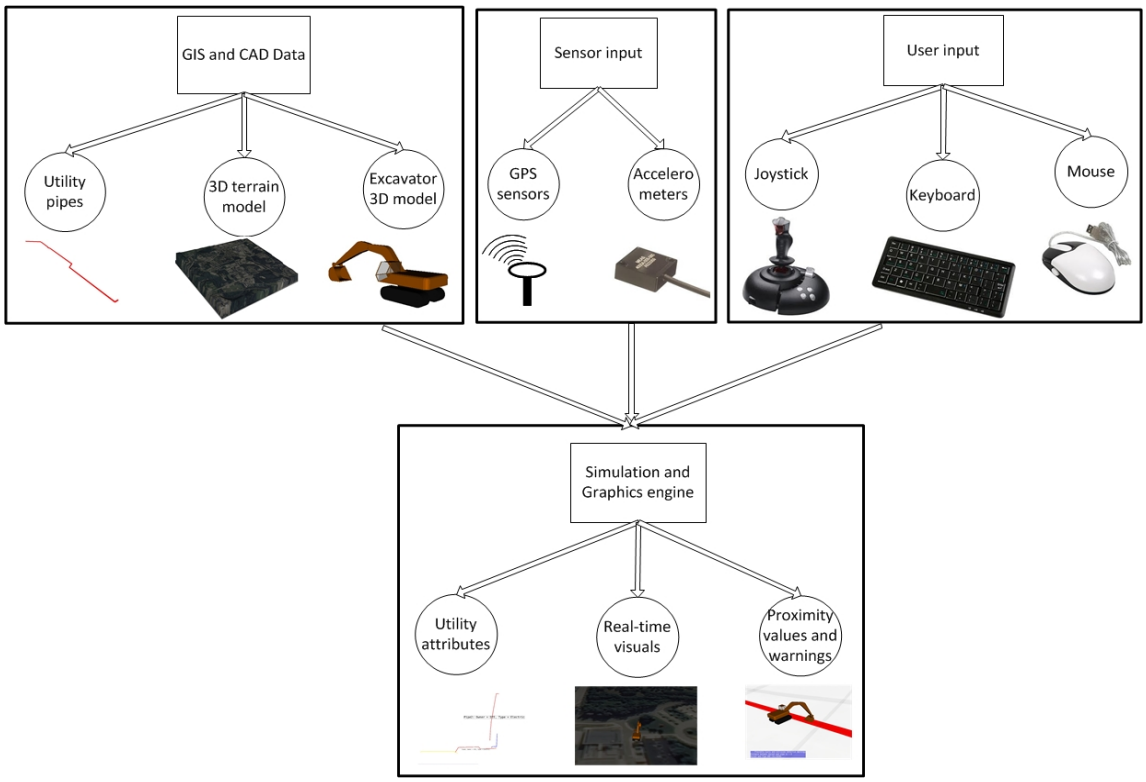

Fig. 2.3: Real-time visual simulation data flow

2.5.3 User Input

An HV simulation uses input from sensors and human interface devices. The fields of teleoperating and mobile remote robotics have demonstrated operators controlling a device present at a remote location through the use of joysticks and other similar means of input (Milgram and Ballantyne 1997). For an HV simulation, the user input completes the simulation loop and is responsible for updating and advancing the HV scene, in particular the equipment models. In the specific case of an HV scene representing an excavation jobsite, the updates to the virtual equipment model will be provided through on-board sensors.

However, this sensor input can be replaced by human interface devices—such as gaming consoles, keyboards, and mice—in order to test the HV simulation during the early stages of scene development, and to test job site safety by simulating a virtual site. The only scenario in which user input is not required to complete the simulation loop is when the HV simulation is used to replay an event using sensor data collected during the course of regular operations. In this scenario, the user input is substituted by the stored data stream that will be responsible for updating the simulation.

2.5.4 GIS and CAD data

An HV simulation requires GIS data, CAD models, or both in order to represent a real-world job site with a convincing amount of detail. A 3D scene must achieve an optimum balance between level of realism and impact on real-time performance. This balance is achieved by creating the 3D models of buildings and utility lines as a wire frame, textured with digital images (Smith and Friedman 2004).

The authors use the term Level of World Representation (LWR) to objectively describe the number of models and their level of detail in a given scene. If an HV scene contains all surface and sub-surface elements of the real world, it is said to have an LWR of 100%. The LWR will largely depend on the type of operation being represented and the goal of the simulation. An eight-parameter index known as the LWR index is introduced in order to categorize 3D scenes into different levels of detail ratings. The eight parameters making up the index are shown in Table 2.1.

Parameter #	Feature	State	Origin	Example
1	Surface	Static	Human-made	Buildings, temporary structures
2	Surface	Static	Natural	Trees, rivers, lakes, soil, rocks
3	Surface	Dynamic	Human-made	Equipment, personnel
4	Sub-surface	Static	Human-made	Utility pipes, conduits, cables
5	Airborne	Dynamic	Human-made	Airplanes, UAVs
6	Sub-surface	Static	Natural	Bed-rock, ground water, sub-surface soil
7	Atmosphere	Static + Dynamic	Natural	Rain, clouds, fog
8	Terrain	Static	Natural	Surface undulation

Table 2.1: Level of World Representation (LWR) for an HV simulation

Parameter 1 includes surface structures such as buildings, temporary structures, and sidewalks. Parameter 2 represents natural features such as trees, rivers, lakes, soil, earth, and surface soil. Any equipment and personnel present on a site are represented by parameter 3. Subsurface infrastructure such as buried utilities, conduits, and transformer boxes is included in parameter 4. Parameter 5 includes airborne elements of a scene such as airplanes and helicopters, if such elements are deemed necessary. Parameter 6 is responsible for the representation of naturally occurring subsurface features such as bed-rock, gravel, earth, and underground water. Parameter 7 is used to represent the jobsite atmosphere such as clouds, rain, fog, and other natural phenomena. The final parameter, 8, represents terrain, including its elevation details and surface imagery. Using these 8

parameters, an HV scene modeler can determine the LWR based on the goals and required output.

In the case of an excavation operation, the goal of the simulation is to assist excavation operators in avoiding underground utility line strikes. In such a simulation, the sub-surface infrastructure is of paramount importance. Only parameters 3, 4, and 8 are modeled in the scene. This yields a 3/8 LWR index score, or a 38% LWR. Thus an HV excavation simulation does not need to have an LWR of 100% in order to be effective. It is essential for the goal of a simulation to be determined beforehand, as modeling real-world elements is a very time-consuming task.

2.5.4.1 GIS Data

GIS has traditionally been used in 2D environment visualization. In recent times, the importance of 3D GIS has been demonstrated in the fields of urban planning, location-based services in shopping and tourism, the pipelines industry, road and highway construction, the telecommunication industry, and environment management (Zlatanova and Stoter 2006). In the case of an HV simulation, GIS data plays an extremely vital role in representing features such as building footprint areas, lot boundaries, and sub-surface utility lines. GIS data consists of two data parts: a geometry part and an associated attribute part. In its current form, the geospatial data is stored in a single object relational database management system (DBMS). The geospatial data is then accessed via a front-end GIS application. The DBMS model allows rapid access to the data and creation of 3D models of the geometry part of the geospatial data (Zlatanova and Stoter 2006). The

geometry part of the geospatial data is typically in a 2D form represented by points, lines, or polygons. Creation of 3D models from geospatial 2D geometry ensures that those models have the same spatial attribute and real-world location as the underlying 2D geometry.

As geospatial data represents features on the earth's surface, their locations in a GIS dataset match the earth's coordinates at those locations. The earth's coordinates at any point on its surface are specified by values of latitude in the north-south direction and longitude in the east-west direction. This reference system is called a Coordinate System. Thus a coordinate system is any fixed reference framework superimposed onto a surface to designate the location of features on or within it. The Earth is roughly spherical in shape, which is more commonly referred to as a spheroid shape. Hence the default coordinate system is the latitude-longitude (and altitude) system referred to as a Spherical or Geographic Coordinate System (GCS), in which points are represented over the 3D surface of the Earth. GCSs are based on an underlying datum that defines the position of the spheroid with respect to the center of the Earth. A change in the datum results in a new GCS and thus new latitude-longitude coordinates for the same point.

While the use of a GCS satisfies the needs of applications representing the Earth's surface in three dimensions, many applications and maps represent the Earth as a flat plane. In order to do so, a new coordinate system called Projected Coordinate System (PCS) must be derived from the spherical system. A PCS is based on an underlying GCS but uses linear units of measure for coordinates rather than the spherical latitude-

longitude. This allows the calculations of distance and area to be done in linear units in a PCS. Furthermore, a GCS is better suited to applications where the Earth is represented as a sphere or globe. In an HV simulation, the terrain would be represented as a plane surface. As a result, a PCS terrain is better suited for an HV simulation.

There are several PCSs, such as the Universal Transverse Mercator (UTM) and State Plane Coordinate System, with each PCS suitable over a certain region of the Earth. The conversion from a 3D sphere in GCS to a 2D plane surface in PCS is achieved by projecting the 3D spheroid onto a 2D plane surface using a map projection. Distortions in shape, distance, and area occur as a result of the map projection. The details of various map projections, as well as their merits and demerits, are described in greater detail in Kennedy (1994). Different PCSs have different units and datum. Due to this, two datasets having dissimilar PCSs cannot be used in the same application unless they are re-projected to a common PCS. Unlike the PCS, the GCS can describe the positions of objects anywhere on the Earth's surface using only their latitude-longitude. Thus the HV simulation would accept input in the form of latitude-longitude, which would be internally converted by the framework to the appropriate PCS. This ensures uniformity of input and accuracy of output.

2.5.4.2 CAD Data

CAD data includes all of the 3D objects that are not part of the GIS database, such as equipment, personnel, buildings, and other temporary structures present on a jobsite. 3D objects can be represented as points, lines, surfaces, or procedural techniques. The use of

points to represent a 3D object is an increasing trend in the field of Computer Vision, where 3D objects often need to be created on the fly using an input of point cloud data. In the fields of 3D simulations, objects have traditionally been represented through surfaces, and to a lesser extent, through solids. For certain visual simulations, the contents of the 3D object are as important as the surface, such as in a medical simulation and terrain erosion representation. These simulations use solid models instead of surface models.

The most commonly used representation for 3D geometry is surface representation of the 3D object. In the surface representation approach, the 3D object's surface is made up of polygons such as triangles and quadrangles (commonly referred to as quads). The individual polygons are bound together to form a polygonal mesh, as seen in Figure 2.4. In HV simulations, CAD objects are required as input for proximity computations and collision detection queries. 3D surface models, which have geometric primitives such as triangles and quads making up their surfaces, can be used as input for these computations and are thus used for representing 3D geometry.

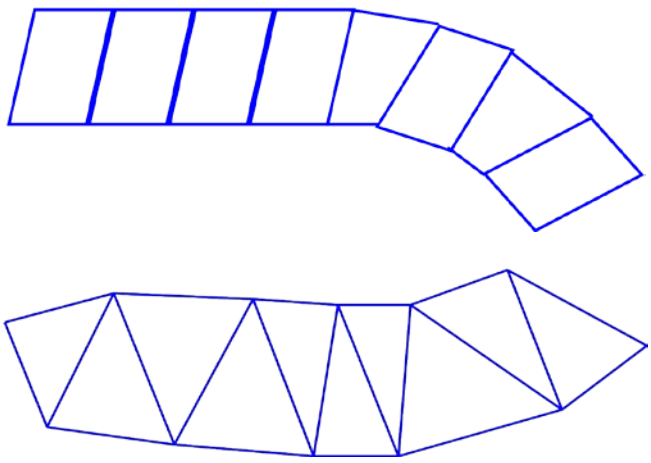

Fig. 2.4: Polygonal Meshes: Quad mesh (above) and Triangle mesh (below)

Within an HV scene, every 3D object appears as an individual entity. However, certain complex 3D objects are built by combining two or more constituent objects. To illustrate this point, consider the example in Figure 2.5 of an excavator machine. It consists of a track at its base and a cabin placed upon the track; the cabin acts as the connection point to the boom, which in turn is connected to the stick, and finally a bucket is connected to the stick. Such a connection mechanism is described as a parent-child relationship.

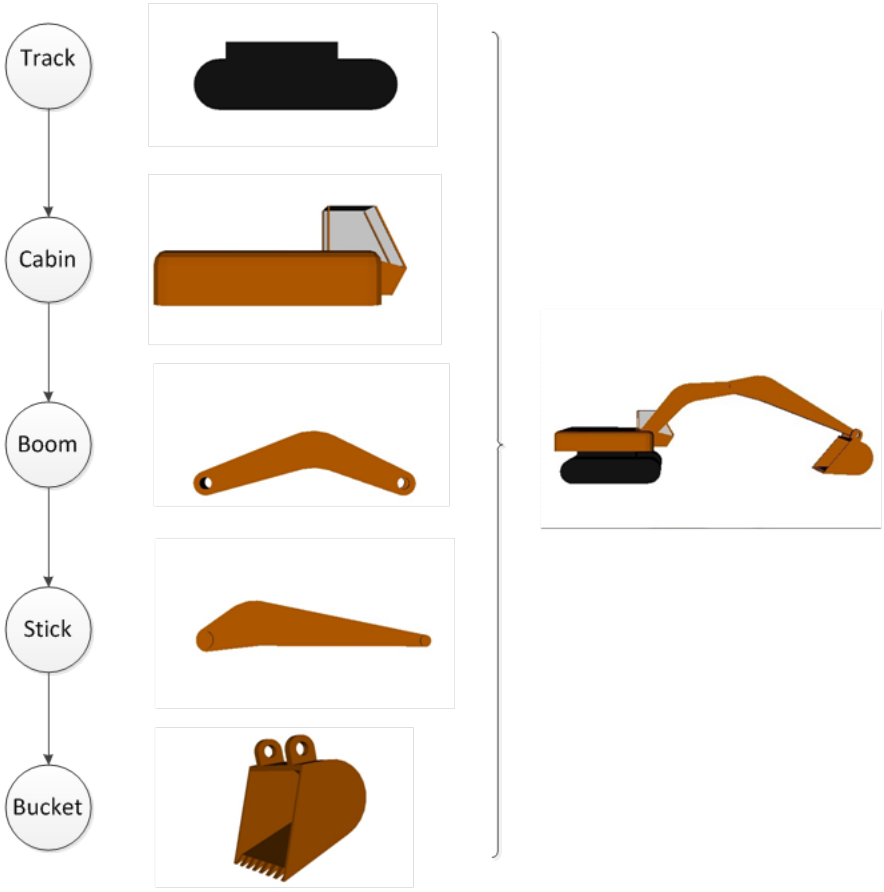

Fig. 2.5: Scene Graph Representation of Articulated 3D object showing construction excavator model

In the case of an excavator, the track is the parent of the cabin, boom, stick, and bucket. A parent object can have one or more children. Certain objects may have no children, such as the bucket object. The parent-child relationship represents the behavior of objects in the real world that have linkages in them. Construction equipment and the human body are two examples of 3D objects in an HV scene whose constituent parts are linked to each other in a parent-child relationship.

When the parent object in such a model undergoes translation or rotation, all of its child objects also undergo the rotation or translation by the same amount as their parent. At the same time, if a child object undergoes a translation or rotation of its own, the parent object's position and orientation remain unchanged. The former point is illustrated by the case of translation of the excavator's track by ten units in the East direction. As a result of this translation, all of the child objects (cabin, boom, stick, and bucket) also move by the same amount in the East direction. However, if the excavator's bucket is rotated by ten degrees away from the cabin, only the bucket object and none of its parent objects are affected. As the parent-child relationship for articulated 3D objects replicates their behavior in the real world, it is an important concept for HV simulations. This parent-child hierarchical representation is also exploited in scene graph layout and structure (Cunningham and Bailey 2001).

2.5.5 Simulation and Graphics engine

The heart of an HV simulation is the Simulation and Graphics engine (SAGE). SAGE receives three things: (1) User Input from human interface devices completing the

simulation loop; (2) Sensor Input from sensors present on equipment, personnel, and structures; and (3) 3D models representing the GIS and CAD data for other objects present on the jobsite. The primary tasks of SAGE are to provide real-time rendering of the HV scene, and to process the multi-source data input to compute the proximities between a pair of objects present in the simulation.

It also provides warnings that may arise out of the proximity computations and displays on-demand attribute display of HV scene elements. The 3D objects in an HV scene are stored in a hierarchical manner. This hierarchy represents their real-world relationships to one another and also leads to efficiency in scene rendering. The authors have identified the scene graph structure as an ideal implementation to achieve these goals.

2.5.5.1 Scene Graph Structure

Computer Science uses graph structures ubiquitously. Of the several types of graph structures, the use of scene graphs in the area of computer graphics has been very successful and widely implemented (Cunningham and Bailey 2001). Scene graphs have a hierarchical tree structure for managing the transformations, level of detail, field-of-view culling, and animation. Scene graph implementations are based upon lower-level graphics Application Programming Interfaces (API) such as OpenGL.

Scene graphs offer a higher-level interface and commands for applications compared to what is possible when directly using a graphics API. In essence, they form another layer of abstraction between the graphics hardware and the application. The scene graph data

structure consists of nodes of different natures. Nodes can be used to store the polygon geometry of 3D objects, apply translation and rotation to objects, act as parent nodes for a group of child nodes, hide and show other nodes, etc. The node at the top of the scene graph is customarily referred to as the ‘root’ node. All of the constituent elements of the scene are child nodes of the root node.

These nodes also have a parent-child relationship that is used to model the 3D objects having linkages, described in the section under CAD Data. Due to their hierarchical nature, scene graph structures allow an entire sub-graph of the overall scene graph from being rendered, thus reducing computational burden. Scene graphs’ ordering and classification of nodes also give the simulation user the option of viewing only certain types of nodes that are of interest at a given instant.

Scene graph structures are not limited in their suitability to the run-time aspect of HV simulations. Their structure also makes articulated object and HV scene creation logical and intuitive. To illustrate this point, we consider the case of a scene modeler whose requirements are the creation of a backhoe excavator, as seen in Figure 2.5. In order to create an articulated 3D object from its constituent parts, the scene modeler must always have the ability to add parent and child nodes in any order to existing nodes. The Scene graph structure allows nodes to be added above (parent) and below (child) a given node in its hierarchy.

The object nodes forming the articulated object may have the requirement to translate, rotate, or both. Transformation nodes can be used to specify the direction and magnitude of rotation and translation of an articulated object's node, as seen in Figure 2.6. For the excavator object being created, the track object must translate as well as rotate in order to replicate the motion of the excavator's track in the real world. When the track node (transform) is rotated, all of its child nodes—cabin, boom, stick, and bucket—also rotate by the same amount. This relationship between transform nodes and their child nodes is represented in Figure 2.6. New scene graphs can be created by combining two or more existing graphs, that is, by adding other scene graphs as child nodes. This enables the creation of an entirely new HV scene from existing scene graph elements.

A scene consisting of a dump truck and excavator on a job site having buried utilities is an example of a typical scene that can be modeled as an HV simulation. SAGE receives input in the form of the 3D terrain model of the jobsite; the GIS data input for the buried electrical conduit and CAD data input for articulated objects represents the dump truck and excavator. The articulated dump truck and excavator are scene graphs, themselves, which can be added to the parent scene graph representing the HV scene.

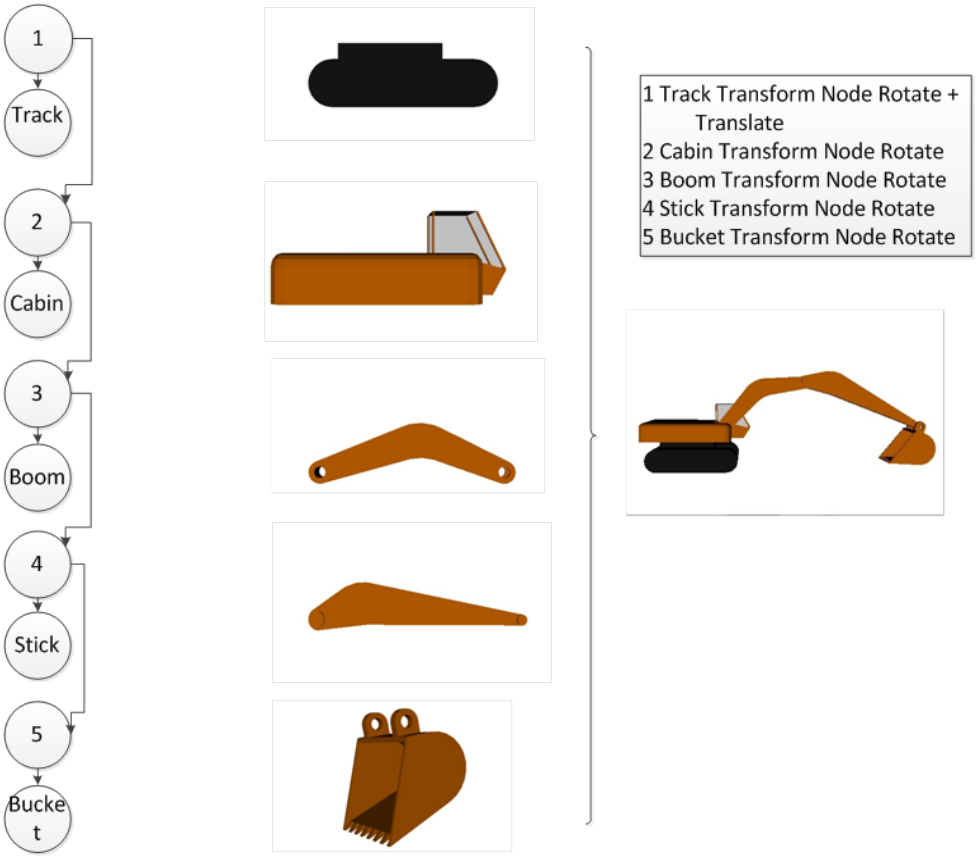

Fig. 2.6: Articulated excavator object scene graph with transformation nodes numbered 1-5

2.5.5.2 Kinematics – User and Sensor Input

For the output of an HV simulation to be useful, all of the entities present in it need to be calibrated to represent their position and orientation in the real world at the start of the simulation. If this is not achieved, the HV simulation will not be an emulation of the real-world job site. This is particularly true of articulated construction equipment due to its dynamic nature. The CAD equipment models loaded into the SAGE are updated with position and orientation data from the job site, which specifies their position and orientation. This task has been successfully included as part of the initialization process

of a visual simulation framework (Stone et al. 1999). After the initialization step, the equipment CAD models have positions and orientations identical to their real-world counterparts.

The transmission of position and orientation data from the equipment to the SAGE is done through the use of sensors and wireless data transmission. The orientation sensors are placed at all joint locations in the equipment that will experience rotation during the course of operation. Position sensors are placed on the equipment body to update their position in the HV simulation. Figure 2.7 is a schematic diagram of a backhoe excavator in elevation and plan, installed with orientation sensors.

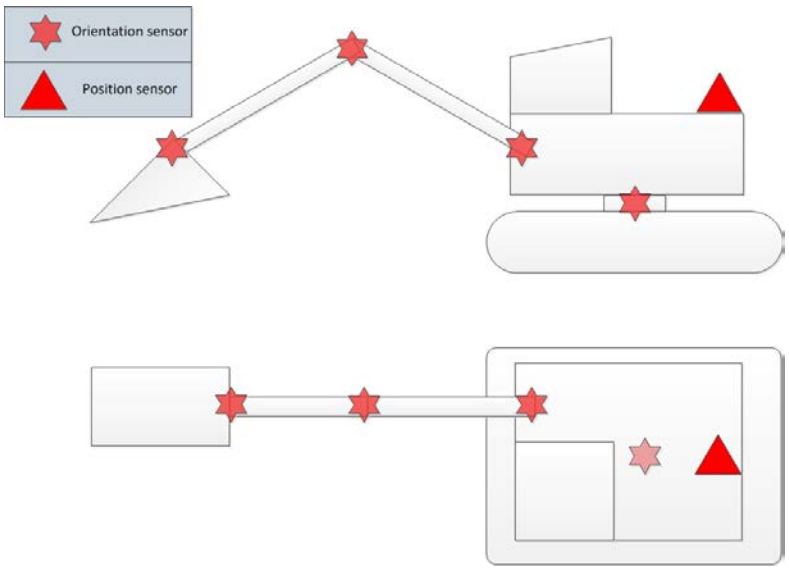

Fig. 2.7: Backhoe excavator with orientation sensors placed to measure changes in angle at every rotational joint

The virtual CAD object representing the real-world backhoe excavator consists of multiple nodes, each having its own transform node, as seen in Figure 2.6. Each sensor on real-world equipment is responsible for updating its corresponding transform in the HV simulation's virtual equipment model. Such an arrangement ensures that every virtual CAD equipment object is an up-to-date representation of the real-world equipment, as depicted in Figure 2.8. Consequently, the warnings from proximity queries and collision detection computations are accurate and serve the aforementioned goals of the HV simulation.

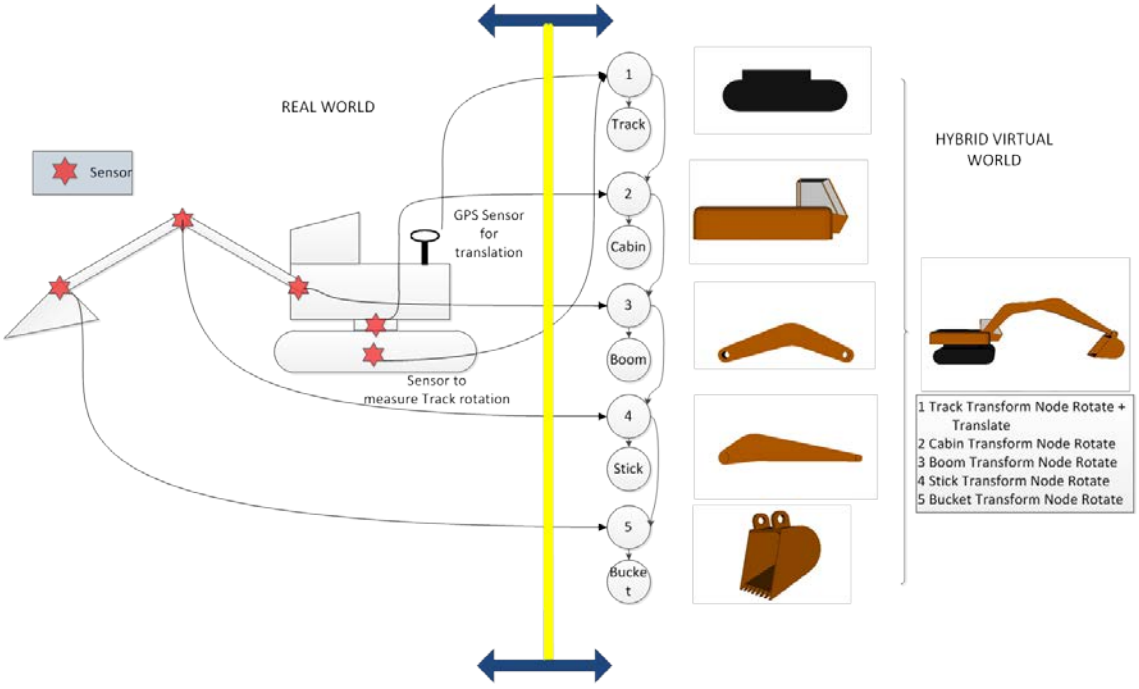

Fig. 2.8: Real-world sensor data updating Hybrid Virtual CAD object

2.5.5.3 Proximity and Collision Computations

Collision detection and proximity queries are a fundamental problem in the field of computer science. Due to their application in computer-aided design and manufacturing,

robotics, simulations, and computer games, this is an area that receives tremendous research attention. In order to better understand the concept of collision detection, it is imperative to know the underlying data for its computation. This section focuses on the application of collision detection and proximity queries to a 3D visualization environment such as HV simulations. The concepts remain largely the same for the other areas utilizing collision detection.

The GIS and CAD data provided as input to SAGE are polygon meshes. Their entire surface is described by a collection of several polygons or geometric primitives. These polygons are either triangles or quads, as seen in Figure 2.9. The figures on the left show the actual 3D models as seen in an HV scene. The figures on the right show the polygonal composition makeup of the surface. The polygons do not have any topological relationship with their neighboring polygons, and the collection of polygons is referred to as a ‘polygon soup.’ The backhoe excavator model in Figure 2.9 consists of 4,982 polygons. A collision is said to occur when polygons belonging to two objects overlap or come in contact with each other. Similarly, a proximity query gives the minimum Euclidean distance between the set of polygons belonging to a pair of objects.

Fig. 2.9: 3D objects (left) and polygonal makeup of surface (right)

Simulation environments requiring proximity information and collision detection have four characteristics. First, models have a very high complexity composed of hundreds of thousands of polygons that together describe the surface of the model. Second, the model's collection of polygons has no topology information and is considered to be a polygon soup. Third, a pair of models can be in very close proximity to each other and share more than a single collision or contact point. Fourth, dynamic visual simulations require extremely accurate output for the proximity and contact between a pair of models (Gottschalk et al. 1996).

HV simulations have all four of these characteristics, which leads to a requirement for a collision detection mechanism that can process data in real time, as well as handle large and complex data sets. As the complexity and size of 3D models increase, so do their number of geometric primitives. It would be almost impossible for a collision detection

algorithm to check for the overlap of every primitive making up a 3D model and have an interactive, real-time performance. To reduce the number of computations, algorithms use the concept of a bounding volume. A bounding volume is created such that it envelops the entire 3D model and all of its primitives.

The 3D model is then recursively subdivided into smaller bounding volumes, as seen in the graphic on the top in Figure 2.10. The subdivision of a model into smaller volumes can be done in multiple ways. The volumes can be constructed as axis-aligned bounding boxes (AABB), bounding spheres (BS), or oriented bounding boxes (OBB). OBBs have been shown to be the most efficient in reducing computation time for scenarios involving large models in close proximity. The bounding boxes form a tree structure, referred to as an OBBTree (Gottschalk et al. 1996). The details of OBBTree building can be found in Gottschalk et al. (1996). The sub-division of 3D models continues until the bounding volumes are enveloping individual geometric primitives, which cannot be further sub-divided.

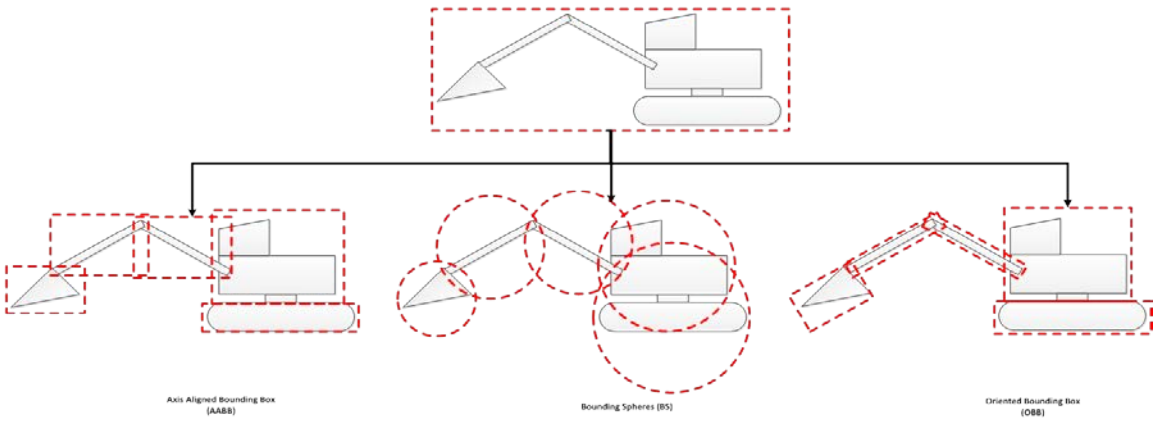

Fig. 2.10: Bounding volume types – (left to right) Axis Aligned Bounding Box (AABB), Bounding Sphere (BS), Oriented Bounding Box (OBB)

To check if a pair of 3D models is colliding, their bounding boxes must be checked for overlap. If the bounding boxes have no overlap, further computations are not required. Thus it is a two-phase approach to determining collisions between models. The two-phase approach saves tremendous computation time compared to a single-phase approach, especially when a pair of CAD models is distant and the possibility of collision is minimal (Kamat and Martinez 2007). If an overlap is found, the next levels of bounding boxes along the OBBTree are checked for overlap (Gottschalk et al. 1996). This is done recursively until the bounding boxes at a particular level do not overlap. When an overlap is found, the detailed computationally intensive test checks for intersections of geometry primitives. This test confirms if a pair of 3D CAD models is colliding and also gives the details, such as the pair of primitives that are involved in the collision.

Proximity queries compute the minimum distance between a pair of 3D CAD models. These queries are similar to collision queries. As the calculation proceeds, the smallest distance between a pair of primitives is stored in memory. The distance value is initialized to a very large value when the query begins. At the end of the query, if this value is found to be zero, the implication is that the objects are colliding. A new category of bounding volumes called Sphere Swept Volumes (SSVs) are found to give superior performance to OBB volumes for proximity queries (Larsen et al. 1999). SSV consist of the core primitive shape that is grown outward by some offset. The offset is referred to as

radius. The shape corresponds to that created by a sphere whose center is moved over all of the points of the core geometric primitive points.

Based on the shape of the underlying geometry, the SSV can be Point Swept Sphere (PSS), Line Swept Sphere (LSS), or Rectangle Swept Sphere (RSS), as shown in Figure 2.11 (Larsen et al. 1999). The bounding volume hierarchy is made up of a tree of SSVs. The tree can consist of a combination of PSS, RSS, or LSS, as determined by the underlying geometry. The proximity query is performed by computing the distance between the primitive core shapes, and subtracting the offset radius of each bounding volume. The details of bounding volume tree traversal can be found in Larsen et al. (1999).

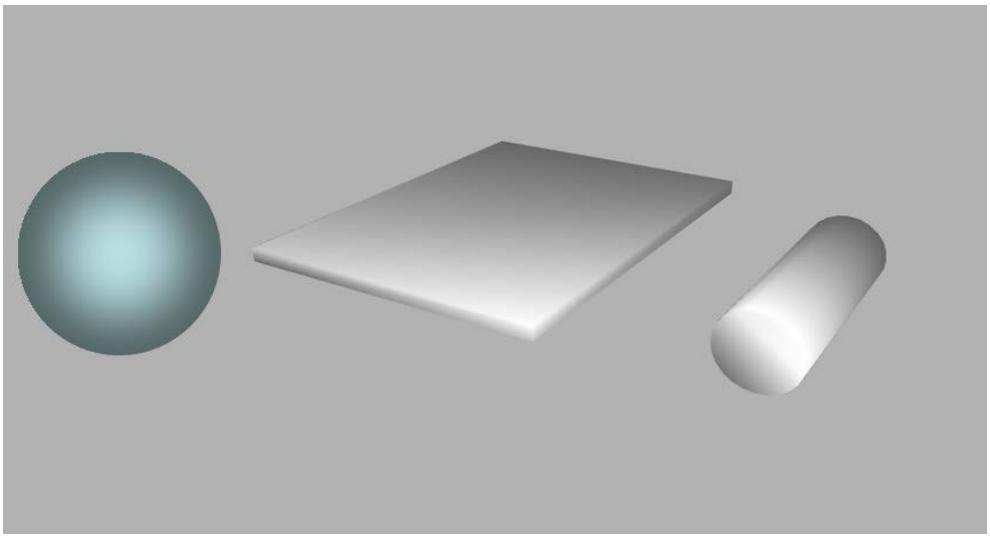

Fig. 2.11: (left to right) Point Swept Sphere (PSS) has geometry primitive point, Rectangle Swept Sphere (RSS) has a geometric primitive rectangle, Line Swept Sphere (LSS) has a geometric primitive of a line (Larsen et al. 1999)

2.6 SeePlusPlus

In this section the authors describe SeePlusPlus, a system designed to allow users to create HV simulations that emulate real-world job sites and their operations. HV scenes can be created using existing sub-graphs of articulated construction equipment, terrain, and buried utilities. SeePlusPlus consists of the following major components: 1) Simulation and graphics engine, 2) Sensor and user input, 3) GIS data, 4) CAD data, and 5) Proximity Queries. An overview of the proposed framework and its constituent components in relation to current implementation of emulating excavation operations is shown in Figure 2.12.

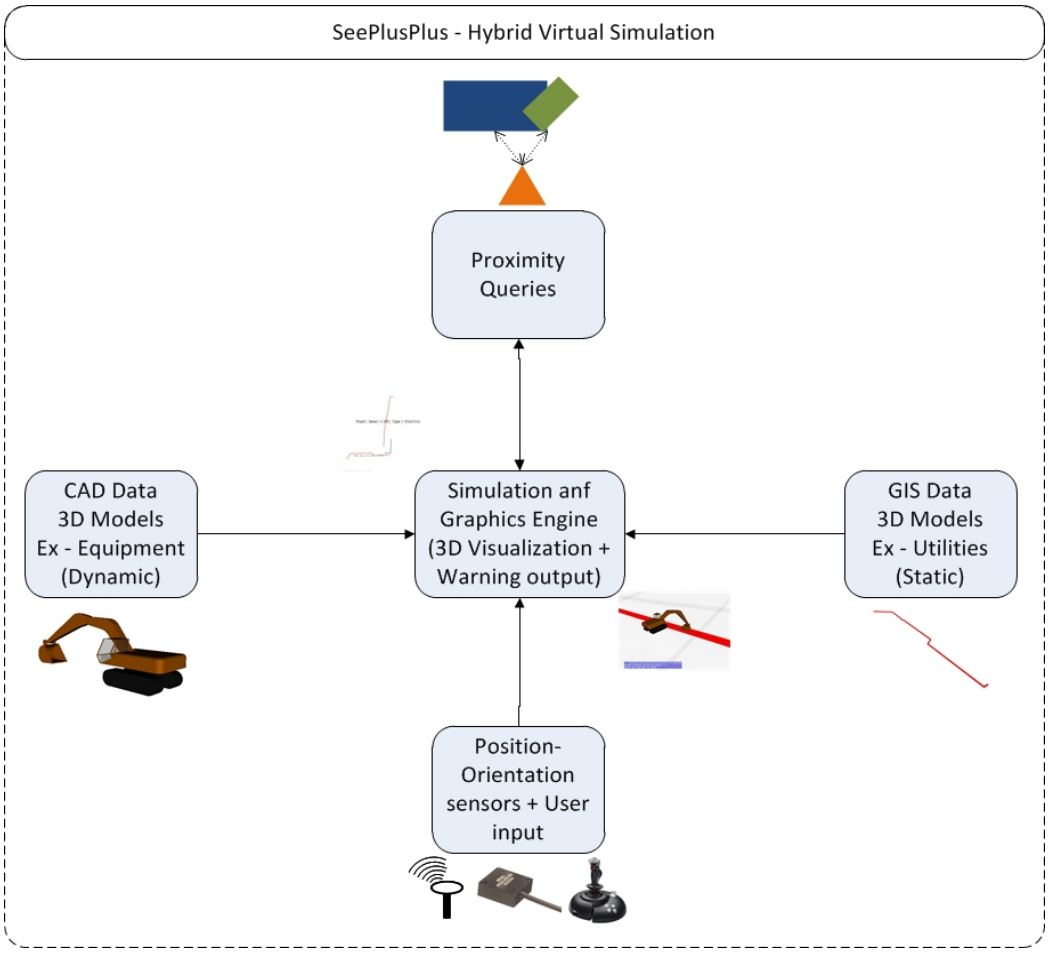

Fig. 2.12: SeePlusPlus overview and components with instantiation examples

SeePlusPlus has been designed using Open Scene Graph (OSG) as its scene graph component of SAGE. OSG is an open-source, cross-platform graphics toolkit based on the OpenGL graphics library. It is implemented entirely in standard C++ (OpenSceneGraph 2007). In addition to the core functionality of a scene graph, OSG has numerous plugins that read/write the most commonly used 2D image and 3D model formats. OSG plugins can load 2D file formats such as jpeg, png, bmp, and tiff, and 3D model formats such as 3ds, ac, and lwo, among others.

OSG's native file format for scene graphs is eponymous and is in human-readable ASCII format, making it suitable for debugging purposes. Run-time applications use faster binary file formats, such as osgb and ive. SAGE is designed to allow the use of both ASCII and binary-based files within the same scene graph. OSG can work with a variety of windowing toolkits, such as Microsoft Foundation Classes (MFC), Qt, and Fast and light toolkit (FLTK). The authors have chosen Qt as the windowing toolkit for SeePlusPlus due its platform-independence.

2.6.1 Articulated object and HV scene creation

A very important goal for the application is to provide users with the ability to create articulated objects using individual components. The user can create parent-child hierarchies and use transform nodes to effect rotation and translation. The OSG framework consists of two Transform node types, namely MatrixTransform (MT) and PositionAttitudeTransform (PAT) Nodes. Transform nodes can have one or more child

nodes and are used as transform nodes for articulated object creation. While creating an object, the user can assign rotation, translation, or both to a Transform node based on the attributes of a given component. For example, the boom's transform node in a backhoe excavator will only be given rotation, as it does not undergo any translation during its operation, unlike the track which undergoes rotation as well as translation.

An HV scene is the aggregation of all articulated object equipment, 3D terrain, and GIS and CAD data that are part of the real world being emulated. Thus a complete HV scene is nothing but a scene graph that uses existing sub-graphs of articulated objects and other 3D objects placed in the correct hierarchical order. The top-most node of the HV scene is customarily referred to as the 'root' node. Figure 2.13 represents an HV scene emulating an excavation job-site and thus has sub-graphs for terrain, articulated excavator object, and 3D models of the sub-surface utility line. The input options for articulated objects, such as the backhoe excavator, are specified at the time of HV scene creation.

2.6.2 Terrain creation

The 3D terrain model is central for the HV simulation in order to represent the real-world job site. Accuracy of the terrain model is directly responsible for the output accuracy of the simulation. The terrain model is similar to the other 3D objects in the HV scene, in that it is made up of geometric primitives such as triangles and quads. 3D Terrain models require the elevation data and a georeferenced image of the terrain. Elevation data is typically stored in a raster format and is a digital representation of the elevation of the

terrain. The most common format for representing terrain elevation is a Digital Elevation Model (DEM).

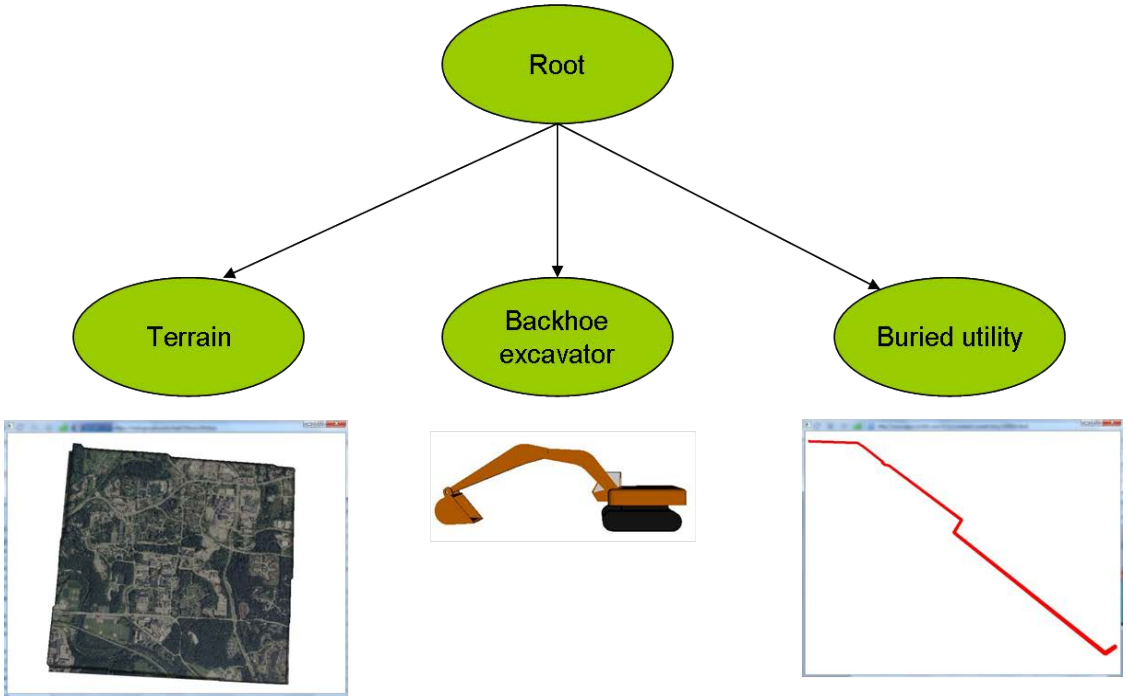

Fig. 2.13: Scene graph for an HV scene representing an excavation jobsite

A DEM refers to a regular grid (raster) of spot heights (USGS 2001). DEMs are basically a sampled array of elevations for the ground position at regularly spaced intervals. The interval spacing defines the resolution of a particular DEM. A smaller-spaced, higher-resolution DEM results in a more accurate terrain model. DEMs are available from a number of sources, such as the United States Geological Survey (USGS). The USGS is the largest repository for elevation data of the entire United States, referred to as the National Elevation Dataset (National Elevation Dataset 2006). The accuracy of the data in the NED ranges from 30 meters to 3 meters for the conterminous United States.

The second requirement for a 3D terrain model is a georeferenced image. The elevation data and image must have the same datum, projection coordinate system, and units for them to coincide. Virtual Planer Builder (Virtual Planet Builder 2011) and osgEarth (osgEarth 2012) are two commonly used 3D terrain creation toolkits. Both of these toolkits have been successfully used to create 3D terrains for SeePlusPlus. VirtualPlanetBuilder (VPB) creates a 3D terrain model file that is stored in memory. This file is a sub-graph that is added to the HV scene when created. OsgEarth, on the other hand, uses an XML file to reference the elevation and image data. The terrain is rendered at runtime and, unlike VPB, the 3D terrain file is not physically present on the hard disk. Both toolkits are based upon the OSG framework and thus seamlessly integrate with any OSG-based application such as SeePlusPlus. The elevation data is processed by the terrain-creating algorithm that creates a 3D mesh representing the real-world terrain using the spot heights from the DEM. The polygon mesh is then draped with the georeferenced image of the terrain to give a complete 3D terrain. Figure 2.14 illustrates the 3D terrain creation process.

2.6.3 3D GIS: Creation of buried pipes

The prevention of excavation-related accidents requires accurate 3D models of buried utilities in order to warn the excavation crews of impending collision. The proximity queries and collision detection computations are performed by algorithms using 3D objects. The utility data is stored in the GIS databases of utility service providers. In some cities and countries, the data for all buried utilities is stored in a single database (Zeiss 2009). The data for utility lines such as pipes and conduits is in the form of lines

and polylines. This 2D data is not representative of the utilities in the real world. Moreover, they cannot be used in collision detection and proximity computations due to the lack of geometric primitives such as triangles and quads. Thus the data from GIS databases or CAD drawings needs to be processed to create accurate 3D models.

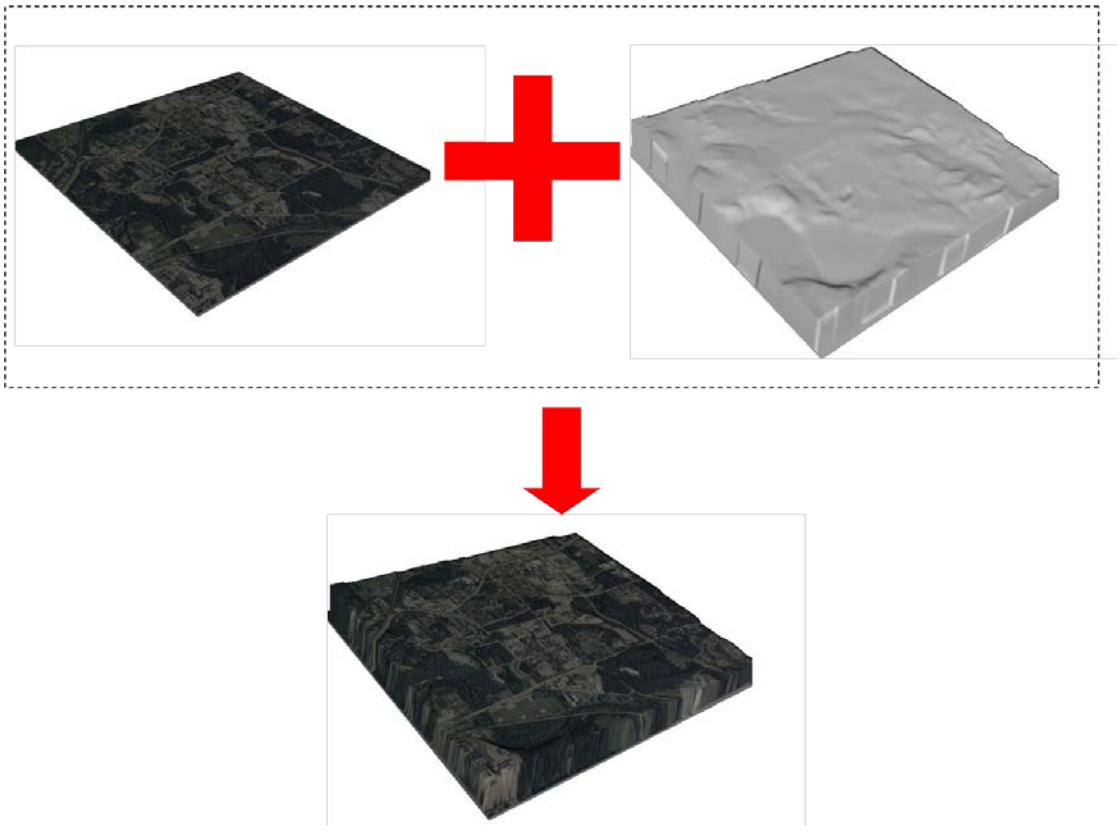

Fig. 2.14: Creation of 3D terrain polygon model using elevation data and georeferenced imagery

The advantage of GIS data is its georeferenced property. Georeferencing is the assignment of a location attribute to information. Georeferencing is essential and necessary in a GIS as all of the information is linked to the Earth's surface. Georeferencing is done through systems such as latitude-longitude, projection coordinate systems, and global positioning systems. Precaution must be taken to ensure that all of

the information being used in an HV simulation is in the identical projected coordinate system and has the same units. Any data having dissimilar georeferencing will not coincide with the rest of the elements in an HV simulation. Thus the utility data from a given database must have the same datum, projection coordinate system, and units as the terrain model being used for the HV simulation. It is observed that the 3D models created from GIS data also show the georeferenced property. This can be seen in Figure 2.15, which illustrates a comparison of the GIS view and the HV environment view of the same buried utility data.

The GIS software used to view and process the data is Quantum GIS (Quantum GIS 2012). Quantum GIS, abbreviated as QGIS, is an open-source, multi-platform GIS software that displays vector and raster data and also has a suite of geoprocessing tools required to create 3D GIS models from the base 2D GIS database. The 3D models' dimensions, such as diameter, are obtained from the attributes of the utility lines in the GIS database. In order to differentiate between different utility types, the 3D geometry is shaded according to the one-call system-marking scheme.

2.6.4 Collision Detection and Proximity Queries

SeePlusPlus' collision detection and proximity query functionality is designed using the Proximity Query Package (PQP) library (PQP Gamma 1999). The PQP library performs computations on 3D objects using their geometric primitives and the object's position-orientation to give extremely accurate output. In order to make the geometric primitives available to the library, the 3D objects must be decomposed into their primitives. These

decompositions are stored in memory as ‘tris’ files. The tris files, along with the position-orientation of the object, are input into the PQP library. The process of decomposition of 3D models into their constituent primitives is achieved using OSG. Each tris file can be used for multiple instances of the same 3D model by substituting each instance’s position-orientation.

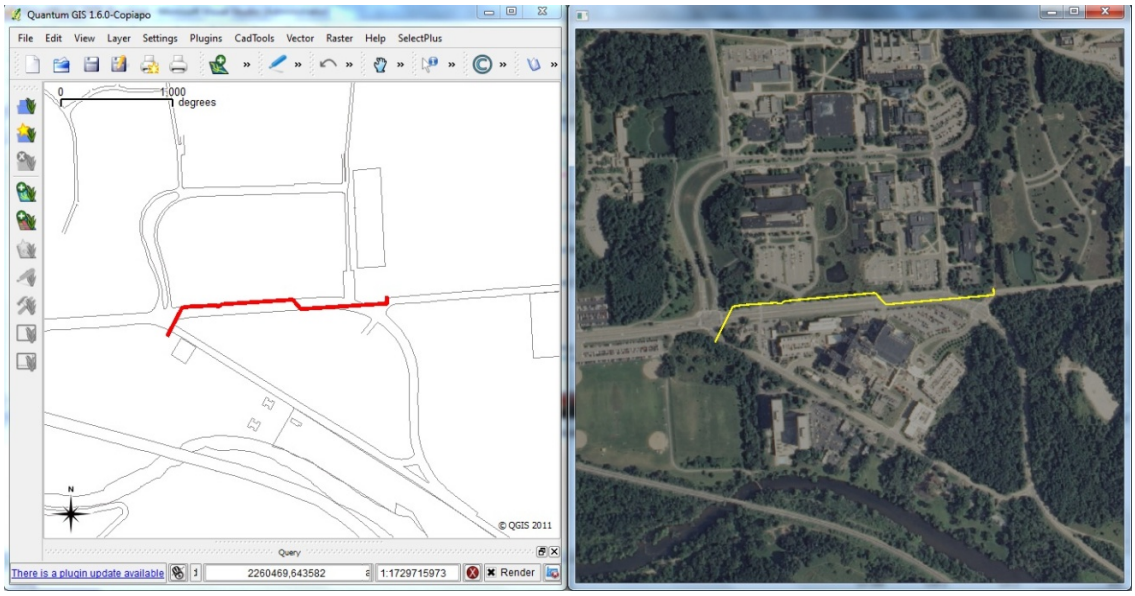

Fig. 2.15: Comparison of screen shots of GIS data viewed in Quantum GIS software and 3D GIS, viewed in SeePlusPlus, demonstrating the georeferencing property of GIS data. Shading is purely for demonstration purposes, as both renderings represent an underground electric conduit.

The collision detection and proximity queries functionality is demonstrated using the example of an excavation jobsite. The jobsite consists of a backhoe excavator and buried electric conduits, the latter being represented by the color red. SeePlusPlus’ SAGE displays the distance of the digging implement to the sub-surface utility in its vicinity.

Thus the excavation crew can take precaution when the digging implement and utilities are in close proximity, potentially avoiding an accident in the process. Figure 2.16 illustrates the output from the PQP computations and 3D visuals from the simulation that display the accuracy of the computations. The excavator is controlled using keyboard input in this simulation example.

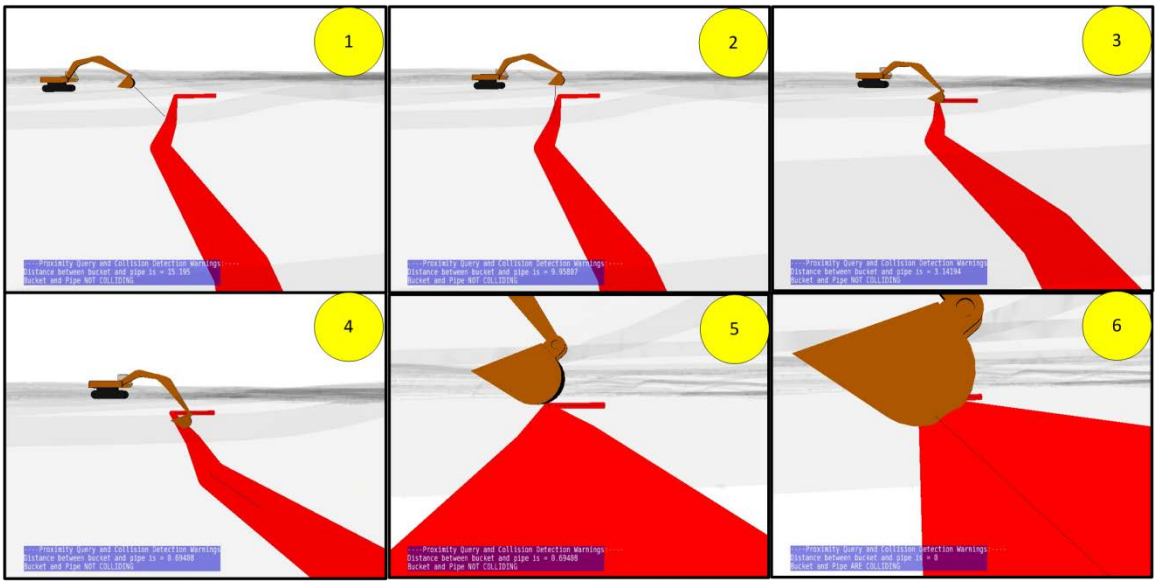

Fig. 2.16: Simulation sequence in ascending order shows excavator moving toward electric utility line and making contact in number 6

2.7 Validation and Results

The authors performed a series of tests to demonstrate the different functionality offered by an HV simulation-

Test 1: To demonstrate the combined sensor input and user input to the SAGE, the following test was performed. The HV environment was designed to emulate the real-

world environment around the Department of Civil and Environmental Engineering at the University of Michigan. A 3D terrain was created using osgEarth from USGS data. The elevation data available had an accuracy of 1/3 arc-second, which equates to 10 meters resolution. The sensor input was from a Garmin eTrex GPS receiver to record the translation motion. User input was through human interface devices to account for rotation motion. Keyboard control was used to rotate the backhoe excavator and move its components, such as the boom, stick, and bucket. The tester was represented by an excavator inside SeePlusPlus, using a combination of User Input and Sensor Input. Thus, in accordance with the HV simulation definition, the tester's translation-rotation in the real world was emulated by the excavator inside SeePlusPlus.

During the test, SeePlusPlus successfully demonstrated the tester's position in the real world inside its 3D visual environment. Thus the test successfully demonstrated the capability of an HV simulation to demonstrate emulation of the real world. During the test, a jittering effect was observed due to inaccuracies in the GPS input. The object appeared to move by small but noticeable amounts, even when the tester in the real world was stationary. This effect can be attributed to the use of a low-cost, low-accuracy GPS receiver. The GPS receiver also displayed a high variability in signal accuracy, ranging from 2 meters at best to 20 meters under tree cover.

Test 2: In order to test the accuracy of the terrain model, a real- to HV-world correlation test was performed. Three locations were selected around the Civil Engineering building, all of which could be easily identified on a map. The first location was the corner of a

curb, the second and third locations were manhole and drain covers, as seen in Figure 2.17. The real-world coordinates of these locations were recorded and used as input for SeePlusPlus. An object was then placed at these exact coordinates in SeePlusPlus. The selected locations are such that any deviation of the object's actual position from its expected position would be easily noticeable. The results from the test are shown in Figure 2.17.

Fig. 2.17: Results from test in top panel, Google Earth views of same points in middle panel, photographs of locations in real world in bottom panel

From the test results, the following observations were made regarding SeePlusPlus and its HV simulation framework. The limiting factor in the accuracy of the framework in the emulation test was the accuracy of the GPS receiver and its unstable output. The real- to HV-world correlation test was affected by the accuracy of the 3D terrain, which is a

direct function of the resolution of the elevation data used to create it. In addition, the ability of the framework to provide warnings and output in real-time is heavily dependent upon sensor stream data transfer speeds together with the processing time required for carrying out proximity queries. For the framework's output to be relevant to a user, it must be provided in time to impact their decision-making process.

The excavator simulation in Figure 16 and Test 1 utilizes user input to simulate orientation updates for the excavator. As the goals of initial tests were to investigate the feasibility of the framework and individual components, orientation sensor input was replaced with human interface device input. Working with a series of orientation sensors placed along articulated construction equipment results in several challenges. The presence of metallic moving parts can affect sensors, such as a magnetic compass, that are impacted by a change in magnetic flux. A compass would be used to provide a value for the heading of the excavator cab. However, the authors are exploring the possibility of using a pair of GPS receivers to provide the heading direction in order to avoid using sensors that would not provide optimum performance when instrumented on an excavator.

The use of accelerometers for obtaining the orientation of articulated joints with respect to horizontal introduces challenges such as latency and resolution. The process of computing the angle of tilt from an accelerometer involves the component of acceleration of the sensor with respect to vertical and horizontal axes. This component is then used to compute the angle that the sensor is making with respect to the horizontal axis. Obtaining

the acceleration values, computing angles, and its subsequent transmission to proximity monitoring components of the framework introduce an added layer of latency or time lag. Another source of uncertainty is the resolution of angle values provided by an accelerometer. A variance of one to two degrees at a given joint can result in a positional variance of several inches at the end-effector when used in a forward kinematic chain. The presence of vibration from the body of the equipment also impacts the output accuracy of the sensors.

The authors plan to account for these uncertainties through the introduction of an uncertainty buffer around the end-effector. This buffer will, in effect, represent a probabilistic zone in which the end-effector can be present. It will also ensure that proximity queries are made in relation to this known uncertainty. In addition to orientation sensors, position data is provided by GPS sensors. GPS has variable accuracy depending upon the clarity of the view that satellites have. The presence of tall buildings in a sensor's vicinity can result in multi-path errors, resulting in a significant drop in accuracy. However, the use of real-time kinematic GPS (RTK-GPS) helps mitigate these errors with resulting accuracy of inches. The implementation of orientation sensors along articulated equipment is part of the authors' future planned experiments, as described in the following section.

2.8 Conclusions and Future Work

The authors introduced the concept of Hybrid Virtual (HV) simulations as a means to emulate real-world operations in a 3D virtual world using tracking and geographic

information from the real world. The various components of the HV simulation framework were introduced and described in detail. The concept of variable degrees real-world representation dependent on simulation goals was also introduced. Initial tests of the HV framework have demonstrated the viability of an HV simulation to emulate real-world operations. The combined sensor input and user input to control a backhoe excavator model demonstrates the potential use of the HV framework to assist excavation crews in preventing buried utility strikes. The tests also showed the limitations of the current framework due to the accuracy of the utility input data, GPS receiver output as well as the resolution of the elevation data affecting the terrain model.

The three main causes of continued unintentional strikes to buried utilities are identified as: (1) the deficiency in accurate position and semantic information of buried utility data, (2) the absence of persistent visual guidance, and (3) the lack of real-time spatial awareness of excavator operators in relation to the proximity of the digging implement to the underlying utilities. The applicability of HV simulation to prevent such excavation-related accidents by alleviating each of the above three problem areas has been outlined.

The future goals are to build a prototype model and carry out tests using sensor input controlling position and attitude of the excavator and track its proximity with respect to sub-surface utilities. Future experiments will include the substitution of keyboard input with orientation sensors such as accelerometers and compasses to demonstrate emulated translation and rotation in a hybrid virtual world. These tests will be carried out using high-accuracy real-time kinematic (RTK) GPS and higher-resolution 3D terrain. This will

allow the team to test the accuracy of the framework without being affected by external data input errors. The team also intends to test the framework's latency and its effect on the ability to provide real-time output and warnings. The latency test will focus on 1) the time delay between any position-orientation change in the real world and its subsequent recording by a sensor, 2) the transmission time between output from a given sensor and its receipt by the framework, and 3) the proximity query and rendering processing time for the framework.

2.9 Acknowledgments

This presented research was partially funded by the US National Science Foundation (NSF) via Grants CMMI-825818 and CMMI-927475. The writers gratefully acknowledge NSF's support. The writers also thank the DTE Energy Company for their support in providing part of the utility data used in this research. Any opinions, findings, conclusions, and recommendations expressed in this chapter are those of the authors and do not necessarily reflect the views of the NSF, DTE, or the University of Michigan.

2.10 References

Beck, A., Cohn, A.G., Parker, J., Boukhelifa, N., Fu, G. (2009) “Seeing the Unseen: delivering integrated underground utility data in the UK”, Proceedings of the GeoWeb conference, Vancouver, July 2009.

Bernold, L.E. (2005) “Accident Prevention Through Equipment Mounted Buried Utility Detection” Trends and Current Best Practices in Construction Safety and Health, Published 31st August, 2005, ISBN 1-886431-09-4.

Call 811 (2010) “John Deere, Common Ground Alliance and US DOT join safe digging”, <http://www.call811.com/for-the-media/news/QuadCitiesOnline.pdf> (06/17/2011).

CGA (2008), Common Ground Alliance, “Call before you dig”, <http://www.call811.com/for-the-media/once-per-minute.aspx> (10/05/2011).

Cunningham, S., and Bailey, M. J. (2001) “Lessons from Scene Graphs: Using Scene Graphs to Teach Hierarchical Modeling,” Computers &Graphics, 2001, number 4.

DoD (1998), “Modeling and Simulation (M&S) Glossary", DoD 5000.59-M, DoD, January 1998 <http://www.dtic.mil/whs/directives/corres/pdf/500059m.pdf> (01/30/12).

Evans, A.J., Roberts, G.W., Dodson, A.H., Cooper, S., Hollands, R., Denby, B., Turner, M., Owen, D., (2003) “Applications of augmented reality: Utility companies”, Survey Review 37: (289) p. 168-176, 2003.

Gottschalk, S., Lin, M. C., Manocha, D. (1996) “Obbtree: A hierarchical structure for rapid interference detection”, In SIGGRAPH ’96 Proc., 1996.

Junghanns, S., Schall, G., Schmalstieg, D. (2008) “Employing location-aware handheld augmented reality to assist utilities field personnel” Lecture Notes in Geoinformation and Cartography, Springer November 2008, <http://www.icg.tugraz.at/Members/schall/lbs_abstract.pdf> (06/20/2011).

Kennedy, M. (1994) “Understanding Map Projections”, ESRI: Manual of ArcGIS.

Kamat, V.R., Martinez, J. (2007), “Interactive collision detection in three-dimensional visualizations of simulated construction operations”, Engineering with Computers, vol. 23, pp. 79-91, 2007.

Korkealaakso, P. M., Rouvinen, A. J., Moisio, S. M., Peusaari, J. K. (2007) “Development of a real-time simulation environment”, Multibody System Dynamics (2007) 17: pp. 177–194.

Larsen, E., Gottschalk, S., Lin, M. C., Manocha, D. (1999) “Fast Proximity Queries with Sphere Swept Volumes”, Available online as technical report TR99-018, Department of Computer Science, UNC Chapel Hill <http://gamma.cs.unc.edu/SSV/ssv.pdf> (06-22-2011).

Milgram, P., Ballantyne, J. (1997) “Real World Teleoperation via Virtual Environment Modeling”, Proceedings of International Conference on Artificial Reality & Tele-existence ICAT'97; Tokyo, Dec. 3-5, 1997.

Miss Utility Call Center Information (2012), <http://www.missutility.net/callcenterinformation/faq.asp> (01/11/2012).

National Elevation Dataset 2006, <http://ned.usgs.gov/> (01/11/2012).

New York Times (1995) “Newark airport is closed as crew cuts power lines”, <http://www.nytimes.com/1995/01/10/nyregion/newark-airport-is-closed-as-crew-cuts-power-lines.html?pagewanted=2> (06/17/2011).

OpenSceneGraph (2007), <http://www.openscenegraph.org/projects/osg/wiki/About/Introduction> (01/11/2012).

OsgEarth (2012), <http://live.osgeo.org/es/overview/osgearth_overview.html> (01/12/2012).

Patel, A., Chasey, A. (2010), “Integrating GPS and laser technology to map underground utilities installed using open trench method”, Construction Research Congress 2010: Innovation for Reshaping Construction Practice [0-7844-1109-3].

PHMSA (2012a), “Significant Pipeline Incidents through 2010 By Cause” <http://primis.phmsa.dot.gov/comm/reports/safety/SigPSIDet_2001_2010_US.html?nocache=9140#all> (01/11/2012).

PHMSA (2012b), “Significant Pipeline Incidents through 2010 By Cause” <http://primis.phmsa.dot.gov/comm/reports/safety/SigPSIDet_2010_2010_US.html?nocache=8248#all> (01/11/2012).

PQP Gamma (1999) “PQP – A Proximity Query Package” <http://gamma.cs.unc.edu/SSV/> (6/22/2011).

Quantum GIS (2012), <http://www.qgis.org/> (01/11/2012).

Roberts, G.W., Meng, X., Taha, A., Motillet, J.P. (2006) “The Location and Positioning of Buried pipes and cables in Built up Areas,” In: Proceedings of the FIG XXIII Congress, Munich, October 2006.

Schall G., Mendez E., Kruijff E., Veas E., Junghanns S., Reitinger B., Schmalstieg D. (2009), “Handheld Augmented Reality for Underground Infrastructure Visualization” In Personal and Ubiquitous Computing, Personal and Ubiquitous Computing. Vol. 13, no. 4, pp. 281-291. May 2009.

Shevlin, T. (2011) “What a Water Main Break Tells Us About the City’s Aging Infrastructure” <http://www.newport-now.com/2011/03/25/what-a-water-main-break-tells-us-about-the-citys-aging-infrstructure/> (06/18/2011).

Smith, G., Friedman J. (2004) “3D GIS: A Technology Whose Time Has Come”, Earth Observation Magazine, November 2004.

Steffen, M. A., Will, J. D., Murakami, N. (2007) “Use of Virtual Reality For Teleoperation of Autonomous Vehicles”, 2007 National Conferences on Undergraduate Research.

Sterling, R. L. (2000), “Utility Locating Technologies: A Summary of responses to a statement of Need Distributed by the Federal Laboratory Consortium for Technology Transfer”, Trenchless Technology Center, Louisiana Tech University, Ruston, LA.

Stone, W., Reed, K., Chang, P., Pfeiffer, L., Jacoff A. (1999) “NIST Research Toward Construction Site Integration and Automation”, Journal of Aerospace Engineering. Vol. 12, no. 2, pp. 50-57. Apr. 1999.

Talmaki, S., Dong, S., and Kamat, V.R. (2010). "Geospatial Databases and Augmented Reality Visualization for Improving Safety in Urban Excavation Operations", Proceedings of the 2010 Construction Research Congress: Innovation for Reshaping Construction Practice, American Society of Civil Engineers, Reston, VA, 91-101.

Thomas, A.M., Rogers, C.D.F., Metje, N., Chapman, D.N. (2007) “A Stakeholder Led Accuracy Assessment System for Utility Location”, 2007 4th International Workshop on Advanced Ground Penetrating Radar : 252-257 2007.

UTNE Reader (2010) “Getting Our Heads in the Sewer”, March-April 2010, <http://www.utne.com/Politics/Water-Infrastructure-Getting-Our-Heads-Sewer.aspx> (06-18-2011).

VirtualPlanetBuilder (2011), <http://www.openscenegraph.org/projects/VirtualPlanetBuilder> (01/11/2012).

Wilder, F. (2010) “The Fire Down Below”, Texas Observer, December 2, 2010 <http://www.texasobserver.org/cover-story/the-fire-down-below> (10/31/2011).

WRAL archives (2007), “Who's to Blame for Cary Gas Line Rupture?” <http://www.wral.com/news/local/story/1916911/> (06/17/2011).

Zeiss, J. (2009) “MEST 2009: Locating Underground Infrastructure in Bahrain” <http://geospatial.blogs.com/geospatial/2009/12/mest-2009-locating-underground-infrastructure-in-bahrain.html> (06/24/2011).

Zlatanova, S., Stoter, J. (2006), “The role of DBMS in the new generation GIS architecture,” in: Rana&Sharma (Eds.) Frontiers of Geographic Information Technology, Springer-Verlag, Berlin Heidelberg ISBN-1- 3-540-25685-7, pp. 155-180, 2006.

Chapter 3

Geometric Modeling of Geospatial Data for Visualization-Assisted Excavation

3.1 Introduction

Beneath the earth's surface there exists an extensive web of cables, wires, pipes, and conduits that transport water, natural gas, electricity, telecommunication services, sewage, and steam. Underground utility lines being struck by mechanized excavators is a long-standing problem. The nation's utility networks are inadvertently struck every minute in the United States (CGA 2008), equating to nearly 500,000 annual utility strikes. The gravity of utility strike accidents is underlined by the congressional Transportation Equity Act for the 21st Century, TEA 21, Title VII, Subtitle C, SEC. 87301, stating that, "…*unintentional damage to underground facilities during excavation is a significant cause of disruptions in telecommunications, water supply, electric power, and other vital public services, such as hospital and air traffic control operations, and is a leading cause of natural gas and hazardous liquid pipeline accidents.*"

Besides the disruptions to public services, daily life, and commerce, utility strike accidents lead to injuries, fatalities, and property damage that cause significant financial loss. For instance, in 1995, a construction crew driving piles for a new parking deck accidently crushed the high-voltage underground electric cables serving Newark International Airport. As a result of the power loss, jet-fuel pumps, escalators, and automatic doors became inoperable. Hundreds of flights were cancelled and thousands of

passengers were left stranded until the cables were fixed (New York Times 1995). In 2007, an excavation strike on a high-pressure gas main in Cary, North Carolina resulted in a 100-foot-high fireball that burned for nearly 6 hours and, consequently, in the evacuation of nearby residents and the closing of major roads (WRAL 2007). In 2010, an excavator hit on an unmarked 14″ gas pipeline near Darrouzett, Texas and killed two workers and seriously injured one (Wilder 2010). The Office of Pipeline Safety's Pipeline and Hazardous Materials Safety Administration (PHMSA)—an agency that maintains data for all reported serious incidents resulting from pipeline strikes—reported a total of 2,770 serious incidents from 2001 to 2010, of which nearly 20% (544 incidents) were excavation-related and caused a total of 37 fatalities, 152 injuries, and $200 million in property damage (PHMSA 2010).

Utility strikes by excavation occur mainly because of the lack of an effective approach to synergize the utility locations and the movement of excavation equipment into a real-time, three-dimensional spatial context that is accessible to excavator operators. Recent research at the University of Michigan created a computational framework to address this excavation safety issue (Talmaki and Kamat 2012). This framework combined sensor-based tracking, georeferenced three-dimensional (3D) visualization, and real-time proximity emulation into a graphical emulation with three interconnected components: 1) geometric modeling of geospatial data; 2) real-time excavator position and orientation tracking; and 3) excavator-utility proximity monitoring (Figure 3.1). A critical lesson learned in this research was the importance of the geospatial utility data and its geometric modeling—the cornerstone to the successful implementation of the framework.

Regardless of sensor precision and computational accuracy of the other framework components, if buried utility data is inaccurate, incomplete, or both, the output from the framework's display and computations could lead to falsely instilled confidence and be counterproductive to the excavator operator.

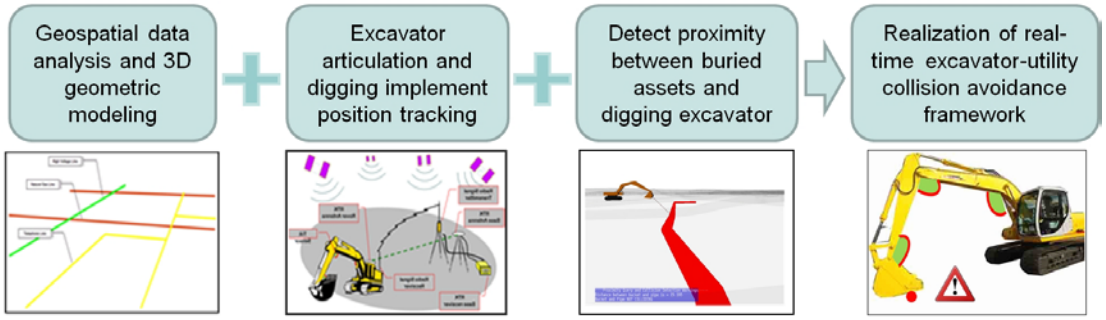

Fig. 3.1: Computational framework for knowledge-based excavation operations

This chapter addresses the computational details in the geometric modeling of geospatial utility data for 3D visualization and proximity monitoring, given its heterogeneity. The remainder of this chapter is organized as follows. The authors analyze the current practice of four key stages in the life-cycle of buried utility geospatial data in Section 3.2. Identified limitations in the aspect of data heterogeneity and ambiguity are presented in Section 3.3. Section 3.4 describes the technical merits of a newly created computational framework in addressing the identified limitations. Section 3.5 presents a toolkit that validates and implements the framework to geometrically represent buried utility geospatial data and the movement of excavation equipment in a 3D emulated environment in real time. Conclusions and future work are provided in Section 3.6.

3.2 Geospatial Data Life-Cycle Management - Literature Review

The life-cycle of the geospatial data of underground utilities typically consists of four distinct phases: data collection, data archival, data updating, and data visualization. The current practices and improvements in each of the four stages are presented in the following subsections.

3.2.1 Geospatial Data Collection

The collection stage of buried utility data is primarily concerned with obtaining the precise location of buried utilities in the horizontal and vertical dimensions. Data collection can occur in two ways depending on the type of operation being performed: 1) installation of new utility lines, and 2) detection of existing lines prior to excavation. When new utility lines are installed, their locations in the field are theoretically reflected in as-built drawings, either digital or paper-based (Bernold 2005b, FHWA 1999). As-built drawings often reference utility locations to existing surface features, such as trees, pavement, and building edges. Other information that is collected at the time of utility installation includes utility dimensions, material, and average depth. Advancements in tracking technologies such as Global Positioning Systems (GPS) and Radio Frequency Identification (RFID) has led to the use of such technologies in collecting accurate utility line locations for new installations in both horizontal and vertical dimensions (Harper and Richardson 2009, North 2010).

Collecting data for existing buried utilities follows a different approach and typically starts from as-built drawings. When data from as-built drawings is incomplete,

inaccurate, out of date, or otherwise undependable, a geophysical survey may be performed to ensure the completeness and accuracy of utility location. Data collected from geophysical surveys corresponds to Quality Level B (QL-B) in accordance with generally accepted definitions of quality levels in Subsurface Utility Engineering (SUE) (Stevens and Anspach 1993; Lew 1996; ASCE 2002). Besides the geophysical survey, non-destructive location techniques and vacuum excavation can be adopted to obtain more precise depths of the buried utilities. Data collected through such techniques represents the highest possible accuracy level, Quality Level A (QL-A), which is also SUE's highest quality level (Stevens and Anspach 1993; Lew 1996; ASCE 2002). The accuracy in the vertical dimension is around 15 mm, and is even better in the horizontal dimension (Jeong et al. 2003).

The locations of buried and newly installed utilities need to be registered to geospatial referencing systems as real-world coordinates to support future analysis and utility retrieval in the field. A number of studies have explored the combination of GPS with precise utility location technologies so that the GPS provides highly accurate real-world coordinates for utility location devices. Consequently, GPS determines the real-world location of utilities based on their relative location to the detecting device. This method is applicable to new utility installations as well as existing exposed utilities (Patel and Chasey 2010). For example, a multi-sensor geophysical survey in combination with high-accuracy real-time kinematic (RTK) GPS was used to collect the horizontal and vertical data of buried utilities in a variety of soil conditions (Griffin 2011). A combination of

GPS tracking and photogrammetry has been proposed for the accurate mapping of buried utilities' locations (Tulloch and Hu 2005).

3.2.2 Geospatial Data Archival

The second stage that buried utility data passes through is Archival. After data has been collected, it needs to be archived in a suitable manner so it can be re-used and updated in the future. Without an effective data archival approach, utility locations that were collected in the data collection stage are either lost or not updated in the archives. The data archival stage is highly complicated due to: the disparate storage methods; heterogeneous data sources; the presence of incomplete, outdated, and inaccurate records; and the merging of the data collection and data archival stages.

Current utility data is stored in both paper and digital formats. In early days prior to the use of computers for digital drawings and inventory, utility locations were typically recorded on paper format, referencing nearby structures such as the edge of some pavement or the corner of a building (Anspach 2011). Since its emergence in the 1970s, the Digital Computer-Aided Design and Drafting (CADD) technology has been steadily replacing paper drawings with digital ones. Metadata—such as the owner of the utility, the utility type, and the date of installation—are also archived in accompanying files for future use. Geographic Information Systems (GIS) is another promising technology for storing utility data mainly because of its capability to store the utility location and attribute data in an integrated manner. GIS allows an owner to have a complete utility inventory stored in a single repository that is easy to update and extract (Corbley 2007).

For utility companies, GIS has become the de facto tool of choice for creating, organizing, and managing geospatial utility information (Sipes 2007). Many utility owners have gone through the transition from paper maps to CADD files or GIS databases via digitization (Cypas et al. 2006), yet quite a few are still using paper drawings. Another format that has been explored for the use of buried utility data storage is Extensible Markup Language, or XML (Cypas et al. 2006). XML is the most commonly used protocol to transfer data for web-based transactions. XML allows users to create their own data tags, which makes it highly suitable to the transmission of buried utility data locations and attributes. Databases allow the storage of XML in its native format.

The increasing use of GPS tracking technology in the field has resulted in a merging of the collection and archival stages. In addition to data gathering, some data collection methods are able to archive the data in one or more of the data archival formats used, such as GIS and CAD (Tulloch and Hu 2005; Hochmuth 2006; Griffin 2011). This combination of collection-archival stages is also seen with the use of Differential GPS (DGPS) receivers to collect the position of the utility, which is then directly archived into maps and resulting in a 1–2 centimeter accuracy (Ordnance Survey 2004).

3.2.3 Geospatial Data Updating

The main concern in the data updating stage is to ensure the currency of geospatial utility data—a critical prerequisite for excavation crews to make safe and well-informed

decisions. The data's currency should be appropriate for the task at hand, and its requirements vary with different user groups (Beck et al 2009). An ideal scenario is knowledge of the most current location information of all buried utilities in vicinity of the excavation job-site. Use of out-of-date maps and drawings has a high probability of leading to an accidental utility strike. Thus data updating represents the most important stage for a damage prevention framework through timely and periodic updates to buried utility data repositories.

Excavation operations represent the most favorable data updating opportunity for all exposed lines to achieve Quality Level A on SUE's Data Quality scale (ASCE 2011). Updates to as-built drawings are made through redlining, an activity performed by utility field inspectors to reflect any changes made to the utilities. With the increasing use of GIS to archive utility data, location updates to GIS databases are also made through mobile GIS units from the field, which typically have the built-in GPS capabilities.

Mobile GIS has been identified as a key technology to keep utility databases up-to-date with an accuracy that is in the range of 10–14 inches (Corbley 2007). This technology enables inspectors who are tasked with verifying the location of pipes and other appurtenances to update the database with the measured locations without first making corrections on a paper drawing and then transferring such corrections to digital databases (Hochmuth 2006). Mobile GIS systems empower crews to update databases with not only the most recent location information, but also photographs and other supporting information. This ensures that the database is always a representation of the most current

field conditions (Bruce 2011). The integration of GPS, mobile GIS, and mobile utility location technologies is enabling data collection, archiving, and updating to take place in near-real-time scenarios.

3.2.4 Geospatial Data Visualization

The oldest form of visualizing buried utility data is through referencing paper as-built drawings. The drawings theoretically represent the position of utilities in the field as they were installed. This method of visualization is used by one-call field-marking personnel to mark utilities with flags, spray paint, and stakes. The one-call utility markings thus represent a basic visualization scheme for guiding excavator operators and crews. With the increasing use of computers, paper has given way to digital CADD drawings and GIS datasets that are visualized in a 2D format.

GIS systems coupled with utility and cable detectors have been used to view buried utility conduits and pipes on portable field monitors (Bruce 2011). Representing the uncertainty associated with a dataset is an important aspect of visualization to enable users to make informed decisions in the field. The use of different levels of blurring and different colors has been attempted to represent the accuracy level of a given utility line in the visualization (Beck et al. 2009). A two-dimensional display of utility data from heterogeneous data sources through a web-based service has been demonstrated and deployed successfully (Beck et al 2009). In recent times, the importance of 3D GIS has been demonstrated in the fields of urban planning, pipelines industry, and road and highway construction (Zlatanova and Stoter 2006). Given depth information, utility data

might be visualized in a 3D GIS environment by vertically offsetting utility lines from their reference surfaces using 3D graphics (e.g. cylinders) to obtain a realistic view in a virtual world. For instance, Huang and Cheng (2009) realized a web-based interactive visualization of 3D pipelines—based on the Ajax3D framework in a client-side browser and the X3D framework—to represent and transport pipeline data.

Another visualization approach has been through the use of Augmented Reality. In this approach, virtual buried utility models are superimposed over the real-world view to give the user a mixed view of the real world as well as virtual underground utility lines. This has been demonstrated to be a potentially useful tool for utility inspectors and maintenance staff (Evans et al. 2003, Junghanns et al. 2008, Schall et al. 2009).

The rapid advancements in the computing power of mobile devices—such as smart phones, personal digital assistants (PDAs), and tablet PCs—empower these devices to run computer graphics applications that were previously only possible on desktop workstations. As a result, the 3D visualization of buried utility networks is possible in the field to assist excavation crews and utility inspectors in making better decisions that can improve excavation safety and productivity.

3.3 Limitations in Current Practice

The current state of practice in each life-cycle stage of underground utility geospatial data is fraught with several limitations that lead to: 1) erroneous utility data with incorrect utility locations, 2) missing depth information and metadata, and 3) heterogeneity in the

data form and accuracy. All three problems prevent the effective use of data in downstream engineering applications, such as information support during excavation. For example, during the installation of a new utility line, changes are often made to the horizontal position and vertical alignment of utilities to account for pre-existing utilities or other obstructions. Such changes are not always recorded on as-built drawings and lead to erroneous data being collected and used in succeeding stages (Patel and Chasey 2010).

Furthermore, the accuracy of as-built drawings in regard to both horizontal and vertical locations of utilities cannot be depended upon reliably as they are approximate in nature (Bernold 2005a). While tracking and locating technologies collect the locations of utilities directly from the field and do not suffer from the limitations of design or as-built drawings, metadata (e.g., equipment accuracy, date of collection, and skill level of personnel operating the equipment) is typically not recorded. The lack of location metadata results in an uncertainty associated with a utility's position that cannot be effectively quantified, represented, or analyzed.

When the accuracy and location uncertainty of as-built drawings cannot be verified, data for existing utilities is collected using geophysical and non-destructive technologies. However, this is an expensive and very time-consuming task. Furthermore, no single technology is capable of detecting all types of materials in wide-ranging soil conditions. To this end, a combination of multi-sensor, multi-frequency location technologies has been suggested as a potential solution for locating and identifying various materials

encountered in variable soil conditions (Sterling 2000). For example, a combination of ground penetrating radar, electromagnetic technologies, and acoustic methods would be more successful at locating buried utilities than each method in isolation would be.

When GPS is combined with a utility detection technology to determine the real-world location of buried utilities, the resulting positional accuracy is typically lower than that of the detecting technology alone; this is because GPS introduces errors, and when considering both GPS and detecting technology errors, the effect is exponential rather than additive due to error propagation issues. For instance, Ground Penetrating Radar (GPR) can achieve a centimeter-level accuracy (Grasmueck and Viggiano 2007), but when fused with GPS, the accuracy is in the region of 0.45m. The use of survey-grade real-time kinematic (RTK) GPS units that provide accuracies in the sub-inch range can limit the error added to collected data.

In regard to data archival, buried utility data has historically been stored in paper-based as-built drawings. Many of these drawings have had utility locations specified in relation to surface features such as trees, buildings, and pavement edges. Any changes to these reference structures outdate the paper archives of utility locations. Prior to the use of CADD and GIS, repair to a damaged pipe required maintenance and construction crews to refer to paper drawings in the storage repository of the concerned utility owner. The lack of digital drawings resulted in a highly time-consuming task of locating the appropriate as-built drawings to extract archived data and obtain utility location information (Corbley 2007).

The utility industry is seeing a shift from paper-based storage media to digital archival techniques such as GIS and CADD. However, data in such modern archival techniques is commonly digitized from as-built paper drawings. This results in the errors, inconsistencies, and uncertainties being carried over from older to newer data repositories, and also results in the introduction of additional errors in the digitizing process. The archival stage also suffers from utility data being present in disparate formats, varying from one utility industry segment to another (Beck et al. 2007). Most of the source data does not store metadata—such as data collection date, owner name, and material properties—in the same repository as geometric representation of the utilities. This results in incomplete archives, and adversely affects downstream utility data stages.

The process of updating geospatial data records is also challenging. The responsibilities of designing, installing new utility lines and repairing them, and servicing existing utility lines are split between different departments. As a result, as-built drawings are created by one department at the time of utility installation, and are then updated through redlining by a different department during maintenance and repair. Such updates are rarely applied to the initial as-built drawings. The multi-responsibility nature of design and maintenance leads to multiple sets of drawings that are out of sync (Corbley 2007). Furthermore, restricted access to competitors' data makes it difficult to update the locations of utilities beyond an organization's own dataset (Beck et al 2009). Thus the lack of effective mechanisms to convert the out-of-date, erroneous, and incomplete data into current,

accurate, and complete datasets can result in the same utilities being wrongly located and marked repeatedly, leading to confusion and misuse during an excavation operation.

The visualization of buried utility data through one-call surface markings has its own distinct limitations, owing to the temporary nature of the markings. Once the top surface has been scraped away by the excavator, the markings are no longer present. If an operator is unable to complete the entire excavation operation in a single uninterrupted shift, the visual guidance is no longer present when they return to finish the task. As a result, the operator must rely on personal judgment and memory rather than objective guidance. The use of paper to represent as-built drawings has a serious limitation, as field personnel need to retrieve utilities on-site based on paper drawings rather than viewing their locations directly. Paper drawings are also prone to degradation over time due to corrections made on the same drawings, and they are expensive to duplicate.

2D visualization platforms are limited due to the cluttered view they present to the viewer while displaying a dense utility network having multiple utility lines stacked over one another. 2D displays also fail to convey depth perception to the viewer. Current 3D visualization in GIS differentiates utilities vertically by referring to the burying depth and the reference surface. But this technique is presently not available to mobile units in the field, and is non-interactive in nature (i.e., excavation crews must subjectively imagine the location and movement of the excavator in the 3D virtual world). This process of imagination occurs in the mind of the excavation crews and is extremely subjective and error-prone.

3.4 Geometric Modeling and Visualization of Geospatial Data - Technical Approach

In order to address limitations inherent in the various life-cycle stages of utility geospatial data, and to mitigate these limitations' negative impact in downstream engineering applications, the authors designed and implemented a scalable visualization framework to characterize and utilize available geospatial utility data for visual information support in urban excavation. A set of specifications was also developed to serve as best-practice guidelines for each stage of the geospatial data life-cycle in regard to the following parameters: Interactivity, Information Richness, Dimensionality, Extensibility, and Accuracy. The newly designed visualization framework was termed IDEAL, an acronym that follows the design parameters (e.g., Interactive, Informative, 3-Dimensional, Accurate, and Extensible).

Any visualization scheme or implementation prescribing to the IDEAL framework is capable of displaying a buried utility network in a 3D environment, complete with any utility attribute information that end-users, such as excavator operators, may require. Any implementation is also extensible to provide additional decision-making support to help end-users complete an operation in an efficient and safe manner. A compliant visualization scheme also characterizes and displays any uncertainty associated with the position of the utilities through color-coded visual buffers surrounding the utility models to aid the end-user's decision-making process. The following sections describe the details and requirements of the design parameters of the IDEAL visualization framework.

3.4.1 Interactivity

A visualization scheme can be categorized as interactive or fixed. An interactive visualization is one that allows users to navigate through it, select an element for closer inspection, highlight areas of interest, or view the scene from different viewpoints (Wiss and Carr 1998). Experiments have shown that users' understanding of a 3D structure improves when they manipulate and interact with the scene (Hubona et al. 1997).

The IDEAL visualization framework requires a prescribing visualization scheme to be interactive in nature, allowing users to view attribute data for any part of the network. The display of on-demand attribute data and user-initiated changes to the displayed elements in the scene make it a requirement for visualization schemes to be rendered in real-time and at interactive frame rates. Studies have shown that a frame rate of 10–15 Hz (frames per second) is adequate for an interactive visualization without having adverse effects on human performance (Chen and Thropp 2007).

Interactivity in the visualization allows users to better perform their tasks. For example, a user interface that toggles transparency of a terrain surface gives the benefit of viewing underground infrastructure when terrain is rendered transparent. At the same time, users can change the transparency so that surface features can be made visible and aid in location on the terrain. Another scenario where user interaction is important is in the selection of a specific utility from a large number of crisscrossed utilities that may be displayed in the scene. The selection would take advantage of attribute data and highlight

the utility that a field engineer may be interested in. The above scenarios are graphically presented in Figure 3.2.

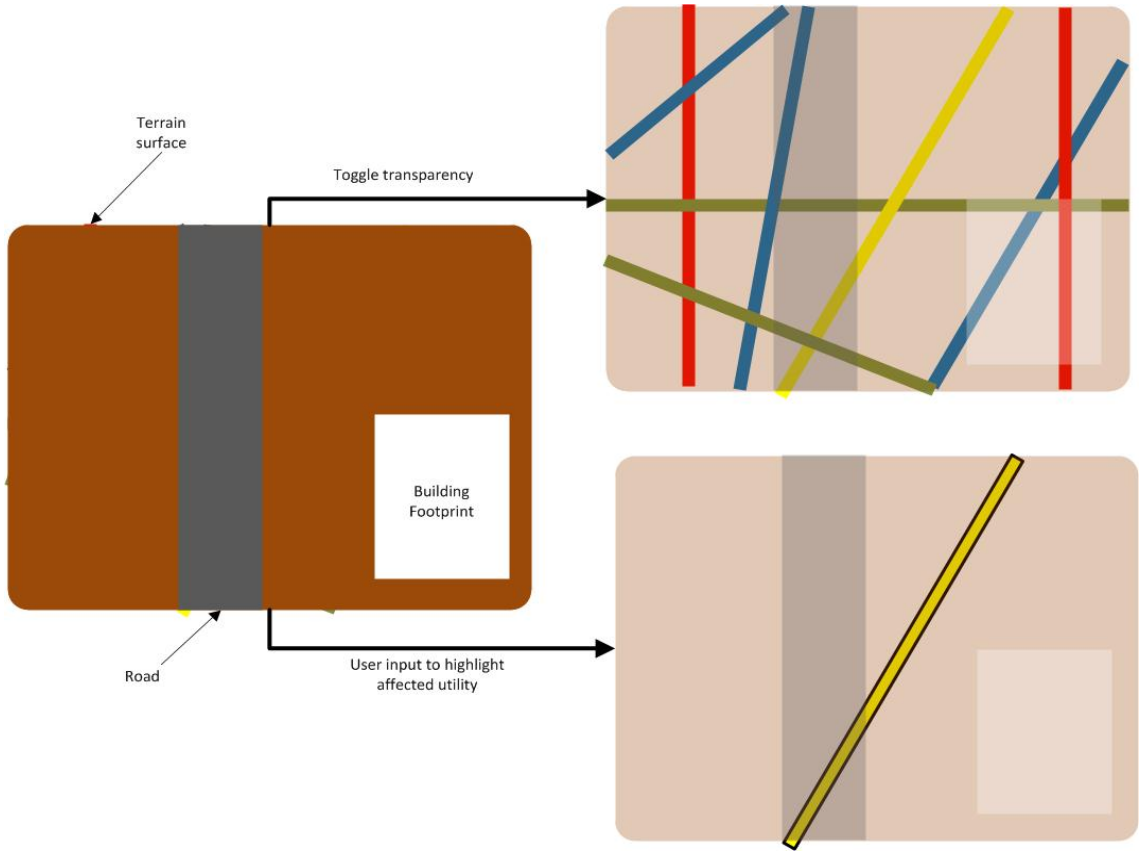

Fig. 3.2: Examples of Interactivity in visualization that increase functionality through user interfaces and input

3.4.2 Information Richness

Conventional visualization through as-built drawings, 2D CADD displays, and mapping software displays represent the utility as a polyline, but do not show the pedigree or lineage of the data. To understand the uncertainty associated with a particular utility line's location, a user has to refer to associated files beyond those used for display. Non-

representation of this uncertainty in the visualization stage can lead to the user operating in a misinformed state. For example, accidents can occur as a result of an excavator operator being unaware of the uncertainties associated with the geodata being visualized.

In order to prevent such a possibility, the IDEAL visualization framework requires the display of utilities to depict not only the location of utilities, but also a region of uncertainty around them. The region can be represented as a “band” or “cylinder” of uncertainty adjacent to the utility line, or as a halo surrounding it, as shown in Figure 3.3.

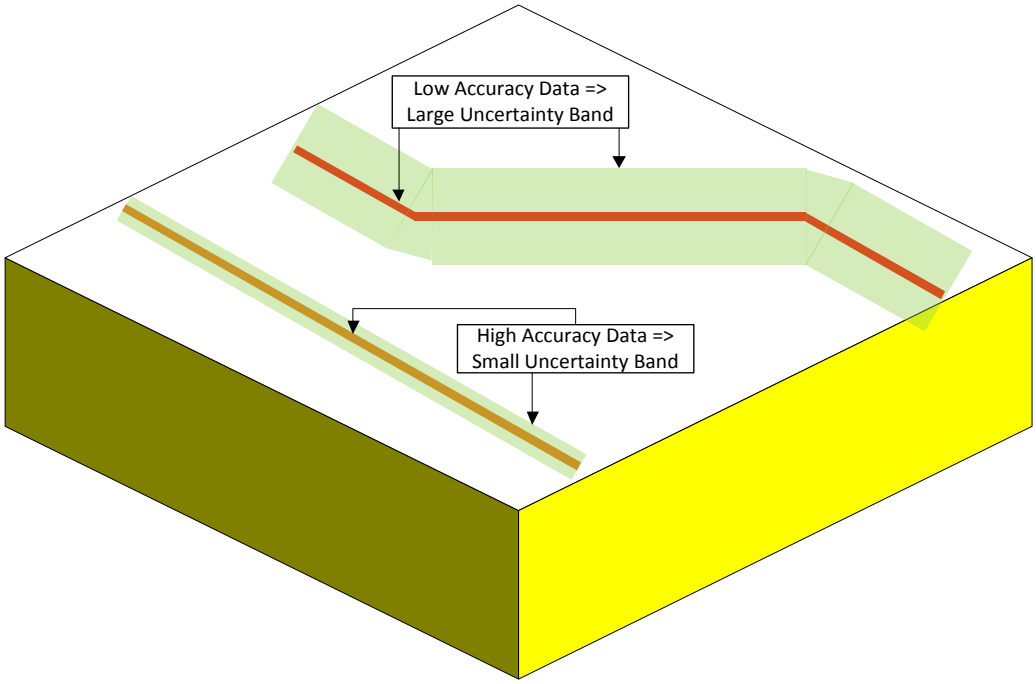

Fig. 3.3: Data uncertainty represented by uncertainty bands to assist excavation crews in making safe and informed decisions in the field

The visualization framework's requirement for displaying any locational uncertainty associated with utilities enables the user to make a decision to proceed with excavation or to require additional non-destructive detection techniques to be used prior to excavation by simply gleaning easy-to-understand information from the visualization.

3.4.3 Dimensionality

An effective visualization framework must be as representative of the real world as possible in order to be useful to the user for their decision-making process. Representing cylindrical conduits and pipes as polylines in a 2D visualization is far from realistic in representing the real world. The IDEAL visualization framework requires 3D models for displaying a utility network. The ability to view utilities in a 3D environment enables users to view a complex network in a clear, unambiguous manner that is not possible in paper drawings, CADD, or GIS 2D visualization. It is a common occurrence for multiple utility lines to be laid in a stacked manner, having the same horizontal location but varying depths, as seen in Figure 3.4.

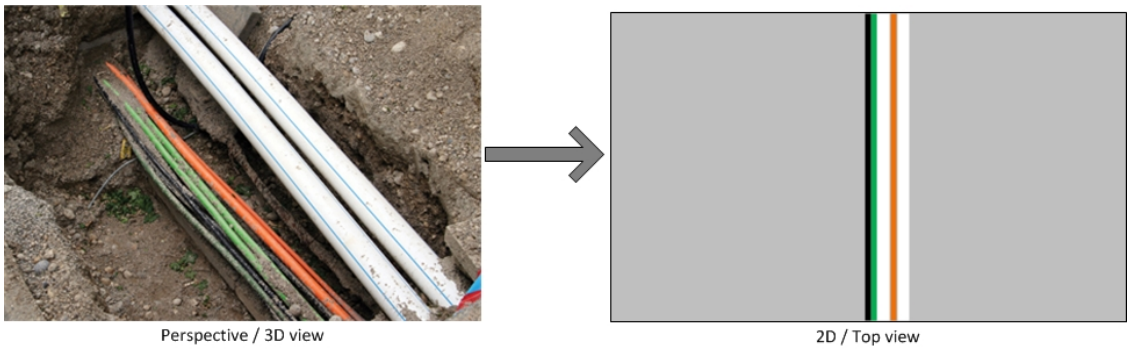

Fig. 3.4: Limitations of 2D visualization for viewing complex, stacked utility networks

Viewing such a complex group in 2D leads to a clustered scene in which individual utility lines and their individual levels of uncertainty overlap. The informative value of a 2D visualization is thus limited owing to the overlap of uncertainty bands and halos, making it difficult for end-users, such as equipment operators, to determine the specific locations and associated uncertainties of specific utility lines.

Visualization of a sub-surface utility network in a 3D environment is inherently linked to the overlying terrain. In existing visualization methods—such as as-built drawings, 2D CADD, and GIS displays—users locate utility networks in relation to existing streets, landmarks, and other known features (Anspach 2011). There is a need for a 3D terrain model to also be present in a 3D visualization framework to effectively emulate the real world. Its absence leads to the lack of a reference point inside the 3D virtual world, as demonstrated in Figure 3.5. The IDEAL visualization framework thus requires a scene to be entirely in 3D and contain the utility network as well as the overlying terrain model.

3.4.4 Extensibility

Visualization is the final stage in excavator operators' use of buried utility geospatial data. In addition to serving the sole purpose of viewing the utility network, an effective visualization scheme is one that can be extended to other uses—such as displaying proximity values and collision warnings to buried utilities—computed by independent

analysis processes. The IDEAL visualization framework thus supports extensibility for processes beyond passive visualization alone.

Fig. 3.5: Comparison of 3D visualization of a utility network (1) with no overlying terrain, (2) with terrain surface, and (3) with translucent terrain (Utility Data Source: Page Tucker, ProStar Systems Inc.)

In the real world, excavator operators are unable to see the underground pipes covered by the earth and soil. They also do not have real-time information regarding the proximity of the digging implement to the underground utilities and thus are not warned of an

impending collision. A 3D visualization in a virtual world is capable of processes beyond those possible in the real world, such as calculating the distance based on the 3D geometric models of the excavator and utilities, and issuing warnings to prevent accidents.

The ability to perform proximity analysis is a particularly useful extension of visualization. The lack of depth information and the inability to see the buried utility can lead an operator into accidently strike the buried structure while performing an excavation operation. The operator can be aided by performing computations using the 3D model of the excavator and those of the subsurface utilities. These computations are based on the use of computer graphics algorithms that utilize 3D geometric models as input. The 3D models can be of a number of types, as shown in Figure 3.6.

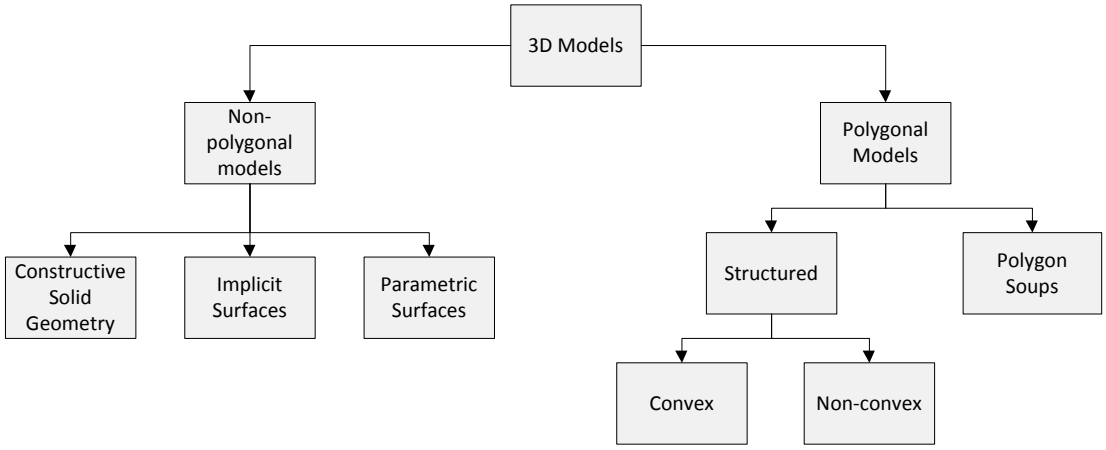

Fig. 3.6: Classification of 3D model types (Lin and Gottschalk 1998)

Polygonal models are the most commonly used type of model in the field of computer graphics and 3D modeling. The polygons composing the model surface are triangles or

quads, as illustrated in Figure 3.7. Their popularity is due to their versatility and ease of rendering (Lin and Gottschalk 1998). Polygon soups are groups of polygons that have no topological relationship with one another; they represent the most general type of polygon models. Most modeling toolkits export 3D models in the form of polygon soups. This results in an even greater use of polygon models as the type of 3D model.

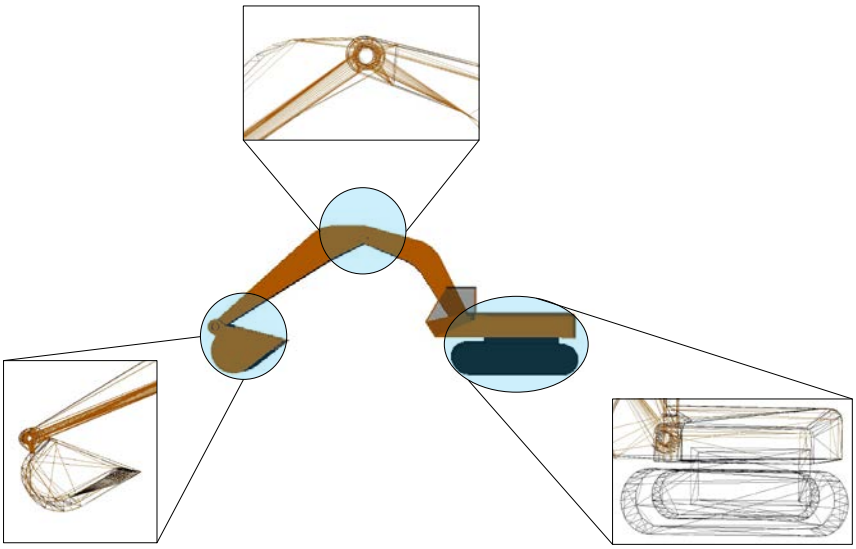

Fig. 3.7: 3D model of an excavator composed of polygon surface in the form of an unstructured polygon soup

An appropriate collision detection algorithm that accepts the chosen 3D model type as input is to be used for proximity and collision computations. The popularity of polygon models for the choice of 3D models leads to a greater number of collision-detection and proximity-monitoring algorithms that depend on polygon models as input data. As the visualization stage utilizes 3D models for the buried utilities, terrain, and excavator equipment, the authors determine that the same 3D model format must be adopted for all

entities present in the 3D scene so that the utility models and equipment models can be used in the same collision-detection algorithm.

The use of dissimilar formats to create the different 3D models can lead to a situation in which the use of dissimilar model types is not possible in the same algorithm. Thus for visualization to be extensible and have greater use beyond the visualization stage, the choice regarding the type of 3D model format must be made with usability in downstream processes in mind.

3.4.5 Accuracy

The accuracy of the visualization of buried infrastructure is of the foremost importance during an excavation operation to aid in the decision-making process. The aspect of accuracy encompasses multiple parameters, of which the positional accuracy of utility data is the most important. The IDEAL framework ensures the positional accuracy of the utility models through the use of georeferenced 3D models. Georeferencing is the process of registering objects based on their location attributes in real-word coordinate systems, for example Geographic Coordinate Systems (GCS) or Projected Coordinate Systems (PCS). Universal Transverse Mercator (UTM) is an example of a PCS. Like all PCSs, UTM provides a constant distance relationship anywhere on the map. In GCSs like latitude and longitude, the distance covered by a degree of longitude differs as you move toward the poles. This makes the use of GCS more suitable for maps representing very large areas. For engineering applications such as utility network representation, a PCS

enables the coordinate-numbering system to be tied directly to a distance-measuring system (MapTools 2012). When buried utility models are georeferenced, they are given an actual physical location that corresponds to their real-world location.

Georeferencing utility location enables the visualization of utilities with other elements—such as a terrain model and real-world data (e.g., GPS sensor input). Without georeferencing utility models, the creator of the virtual scene must manually specify the location and orientation of each utility element, and adjust its display scale during every interaction of the scene. Thus georeferenced buried utility 3D models are essential to ensuring the feasibility and accuracy of the visualization. The IDEAL framework requires the use of a common coordinate system with identical units for storing or converting all elements of a scene, such as utilities, terrain, and sensors.

The dimensions and features of the utility models constitute the second area that affects the visualization accuracy. To be accurately visualized, the 3D utility models need to represent real-world utilities in terms of shape and size. Shape is characterized by the cross section type. The authors determined the requirement of a 3D modeling framework to be able to accept input data of any shape to produce identical 3D models. The commonly observed shapes for utilities are circular and quadrangular (Figure 3.8).

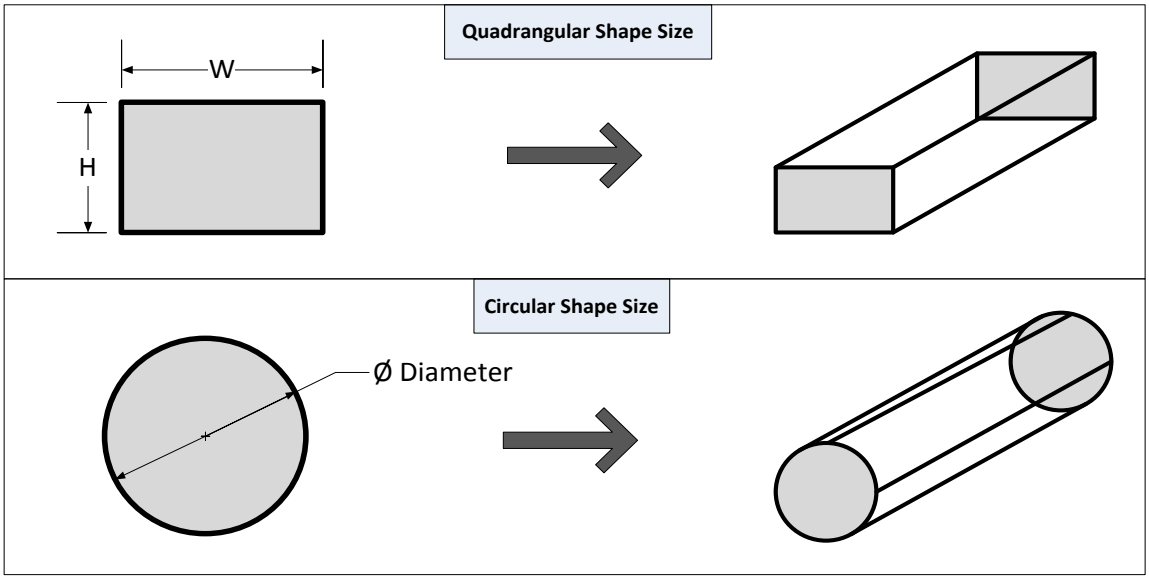

Fig. 3.8: Shape and Size of utility data is required to create 3D models that are representative of the real world

Size refers to the diameter (or breadth) and height of the utility cross section. The size data together with shape data is required to determine the cross section information of the utility. Both shape and size data are archived in GIS databases as attribute data. In open data transmission protocols such as XML, the shape and size data can be captured using specific data tags.

In addition to shape and size data, the authors have determined the need for an unambiguous approach for specifying how the utility location data is collected and archived. During the data collection stage, location data is collected at every point in which the direction of the utility line changes, as illustrated in Figure 3.9. Using this

approach, it is safe to assume that the utility line is a straight line between two successive points when viewed in a 2-dimensional planar view.

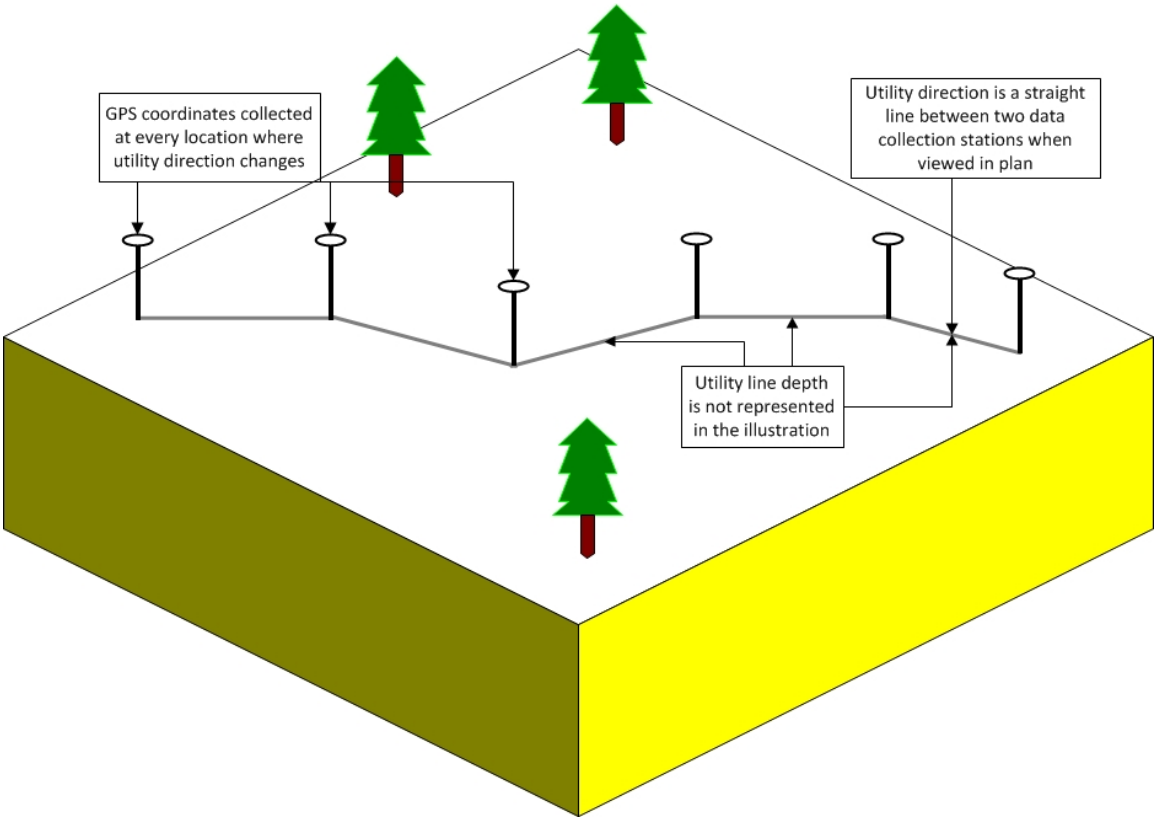

Fig. 3.9: Utility data collection method to facilitate 3D modeling of buried utilities

Location data collected at every turning point consists of three elements: latitude for horizontal location, longitude for horizontal location, and altitude for vertical location. In the case of buried utility data collection, the altitude element can be the source of much uncertainty and error. The elevation data collected can refer to: the elevation of the ground surface; the elevation of the top, middle, or bottom of the utility; or any random location on the utility (Figure 3.10). The specific location on the utility structure at which the elevation is obtained must be recorded.

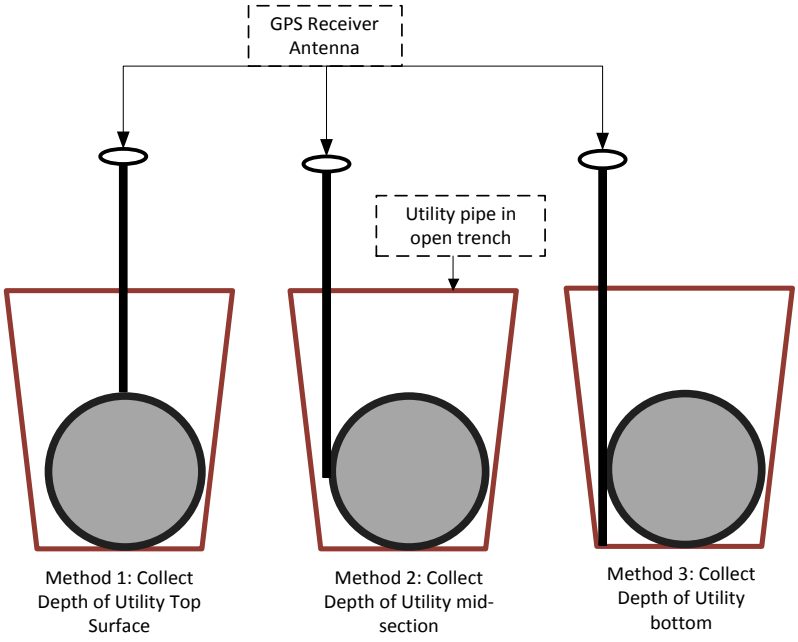

Fig. 3.10: Archival of utility depth collection method is vital for creation of accurate 3D utility models

The authors determine that recording the elevation of the top surface of the utility structure after it has been placed would lead to the least uncertainty and error. As this method would require placing the GPS receiver pole on the utility crown, it does not need addition/subtraction of a horizontal offset to correspond to the utility centerline. In addition, this method directly measures the elevation of the utility (top) surface that is most likely to be struck by an excavator, and hence it is directly applicable to the deduction of safe excavation depths. The importance of an accurate and unambiguous approach for specifying how utility data was collected and archived in achieving an accurate visualization is clearly evident. A 3D modeling framework must take into account any horizontal or vertical offsets, as well as the type of location data being

supplied to it, to create 3D utility models that represent the real-world utilities most accurately in terms of horizontal and vertical location, shape, and size.

3.5 Validation and Implementation

The authors implemented the IDEAL framework through the creation of a visualization tool. In addition, the authors' handling of buried utility data flow through its various stages is described in the following sections, ultimately leading to the creation of 3D models that can be used in IDEAL-based visualization.

3.5.1 Buried utility data flow

In this section, the authors describe their implementation of buried utility data collection, and archival, both for use in 3D modeling. The utility data collection techniques described in Figures 3.8 and 3.9 were implemented using an interface that enable a user to store streaming GPS data to represent a series of utility vertex locations. This graphical user interface also allows a user to select the utility type, its cross section shape, and dimensions, as shown in Figure 3.11. This tool was used to collect data for a collection of hypothetical utilities around the Department of Civil and Environmental Engineering at the University of Michigan.

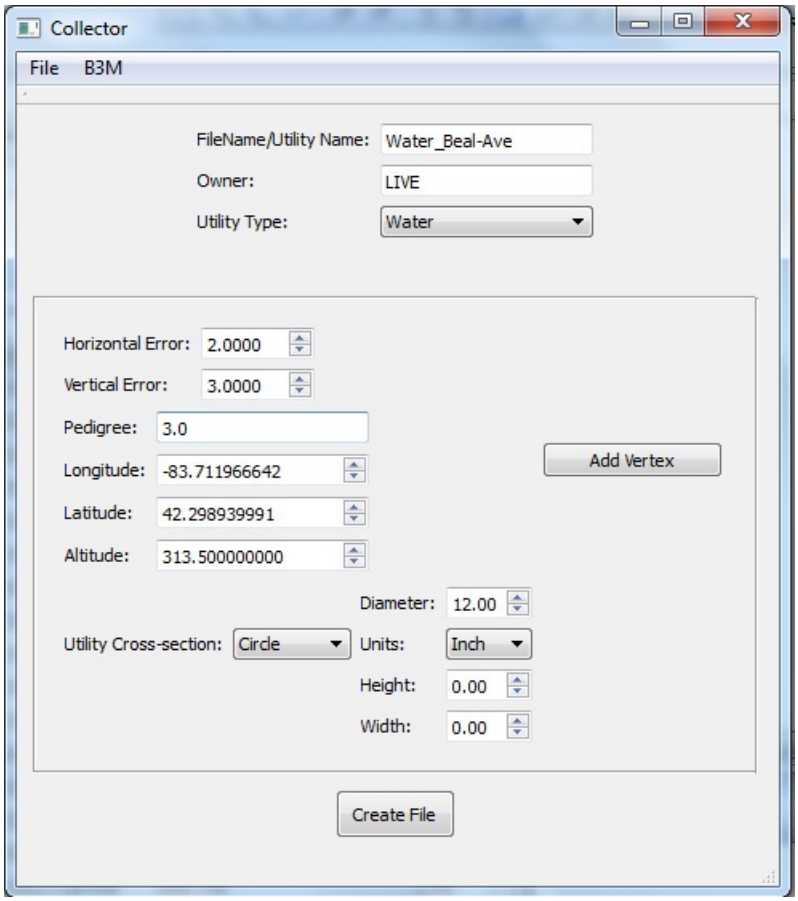

Fig. 3.11: GUI-based tool for collecting buried utility data

Data is archived in an XML file when saved from the data collector GUI. The XML files are subsequently used to create 3D models of buried utilities. Thus the hypothetical buried utility data passes through stages of collection, archival, and visualization. The scenario would be identical for an actual utility whose location and elevation would be recorded appropriately, as shown Figure 3.10. Thus the data flows through the three stages, ultimately leading to a 3D visualization that implements the IDEAL requirements. As the data was collected for hypothetical utilities, no update stage was required. The data flow adopted for buried utility data is shown graphically in Figure 3.12.

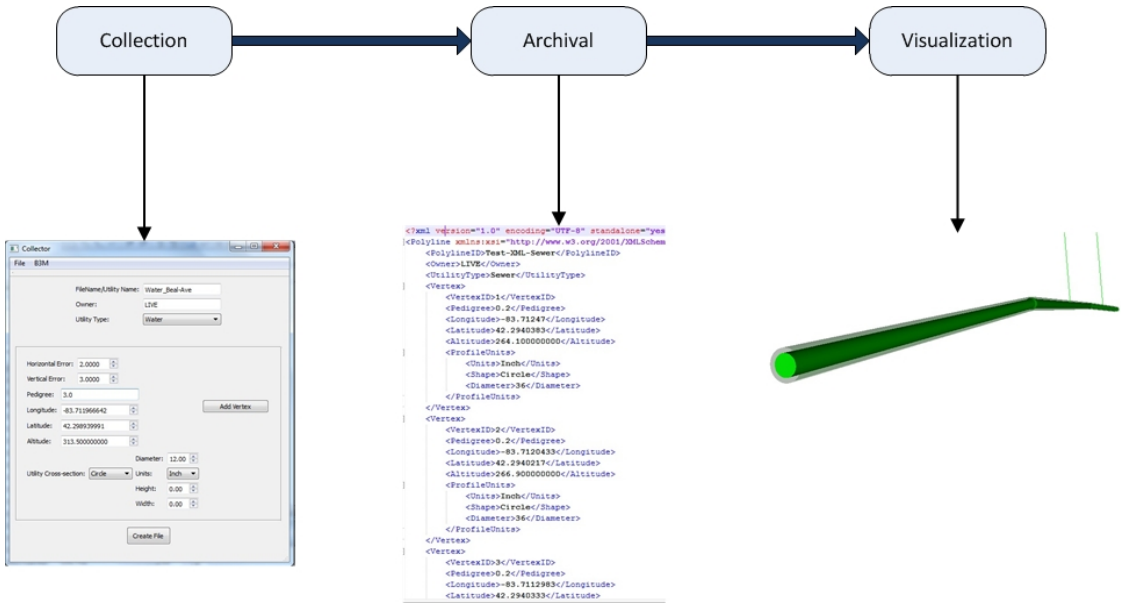

Fig. 3.12 Buried utility data flow through collection, archival, and visualization stages

3.5.2 Visualization

A visualization tool for buried utility maintenance, excavation, and other engineering applications requires input from various data sources and follows a sequence of processes, as shown in Figure 3.13. Such a visualization conforms to the IDEAL framework when there is: a provision for user input and interaction; use of 3D models to represent the utility network and other scene components such as terrain and excavator; and an extensible component such as proximity monitoring that takes input from real-world sensors.

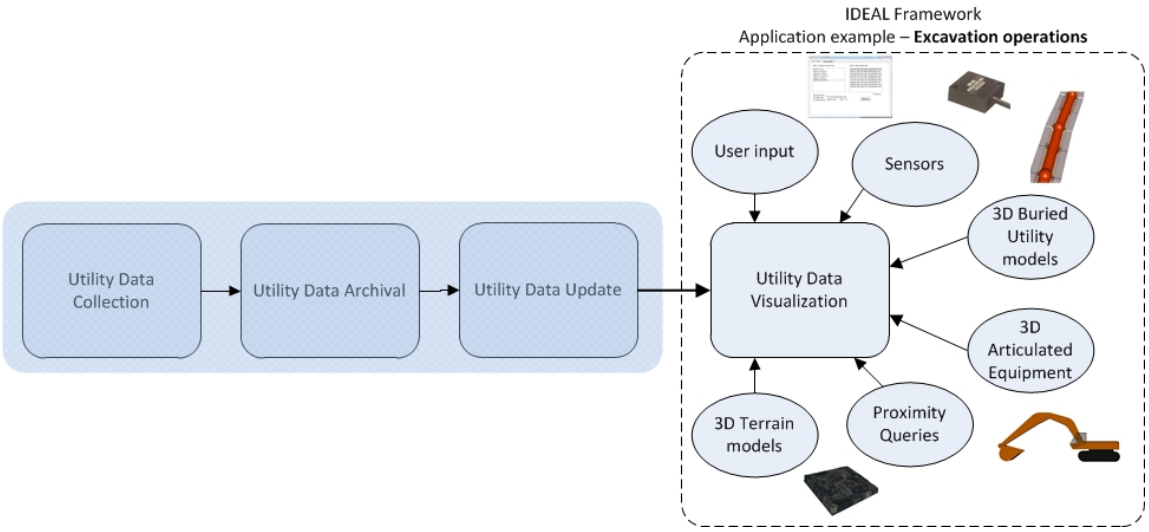

Fig. 3.13: Buried utility visualization conforming to IDEAL framework

The authors implemented a buried utility visualization and analysis environment based on the IDEAL framework for safe urban excavation operations. The visualization environment is based on scene graph architecture, with Open Scene Graph (OSG) being the specific scene graph implementation used. OSG is an open-source cross-platform graphics toolkit based on the OpenGL graphics library. It is implemented entirely in standard C++ (OpenSceneGraph 2007a). OSG-based applications are capable of handling a wide range of 3D model formats, and thus the use of 3D geometry in the polygon surface format is also ensured. Scene graphs have a hierarchical tree structure for managing the level of detail, field-of-view, culling, transformations, and animation. These features allow user interaction through appropriate interfaces to customize the visualization. Scene graph implementations are based upon the lower-level graphics application programming interfaces (APIs), such as OpenGL. Thus the use of OSG for seamless 3D visualization helps meet the Dimensionality requirements of the IDEAL

framework. The visualization's graphical user interface component is developed using the cross-platform tool, Qt. Figure 3.14 shows multiple views of a 3D environment through variable transparency and viewing orientations. The user interface also enables users to create proximity queries between an excavator and underground utilities to provide collision warnings.

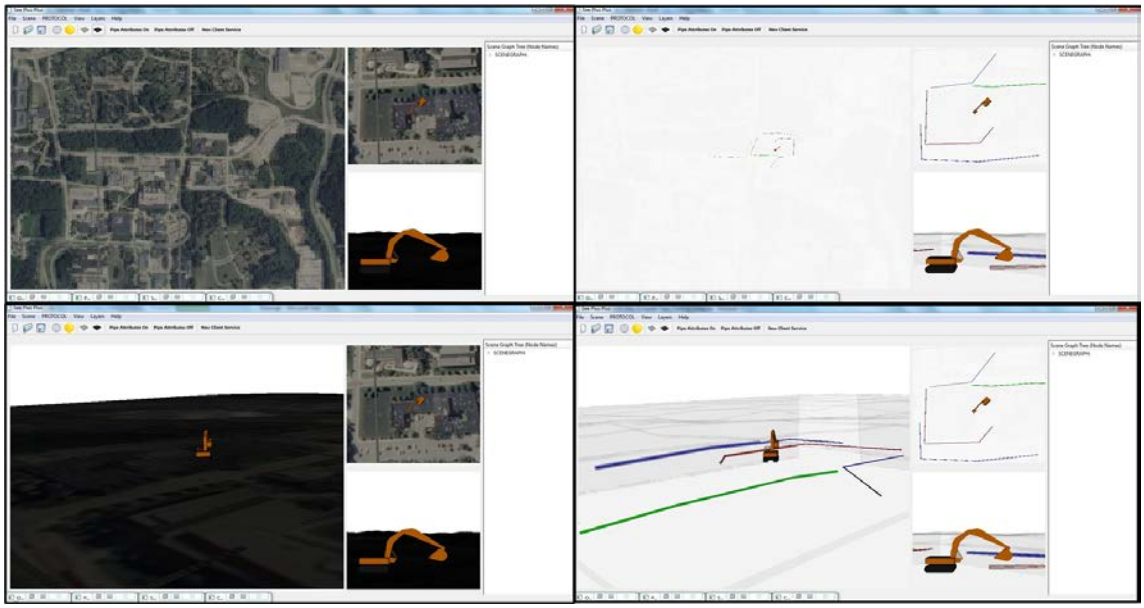

Fig. 3.14: User interactivity in visualization through changing of terrain transparency and viewing perspective

Effective visualization also prevents information overload and gives users the option to view only those parts of the scene that may interest them. The user interface implementation allows users to view a scene in a manner that best suits their requirements. For example, the visualization user interaction also allows users to choose the level of information they wish to see by turning on/off various layers (e.g., terrain)

and toggling on/off utility attributes, as shown in Figure 3.15. Furthermore, through the visualization of the uncertainty buffer around a utility, as well as its proximity to surrounding utilities, field engineers can decide whether to proceed directly with a mechanical excavator or to employ additional precautions such as non-destructive testing and/or hand excavation methods.

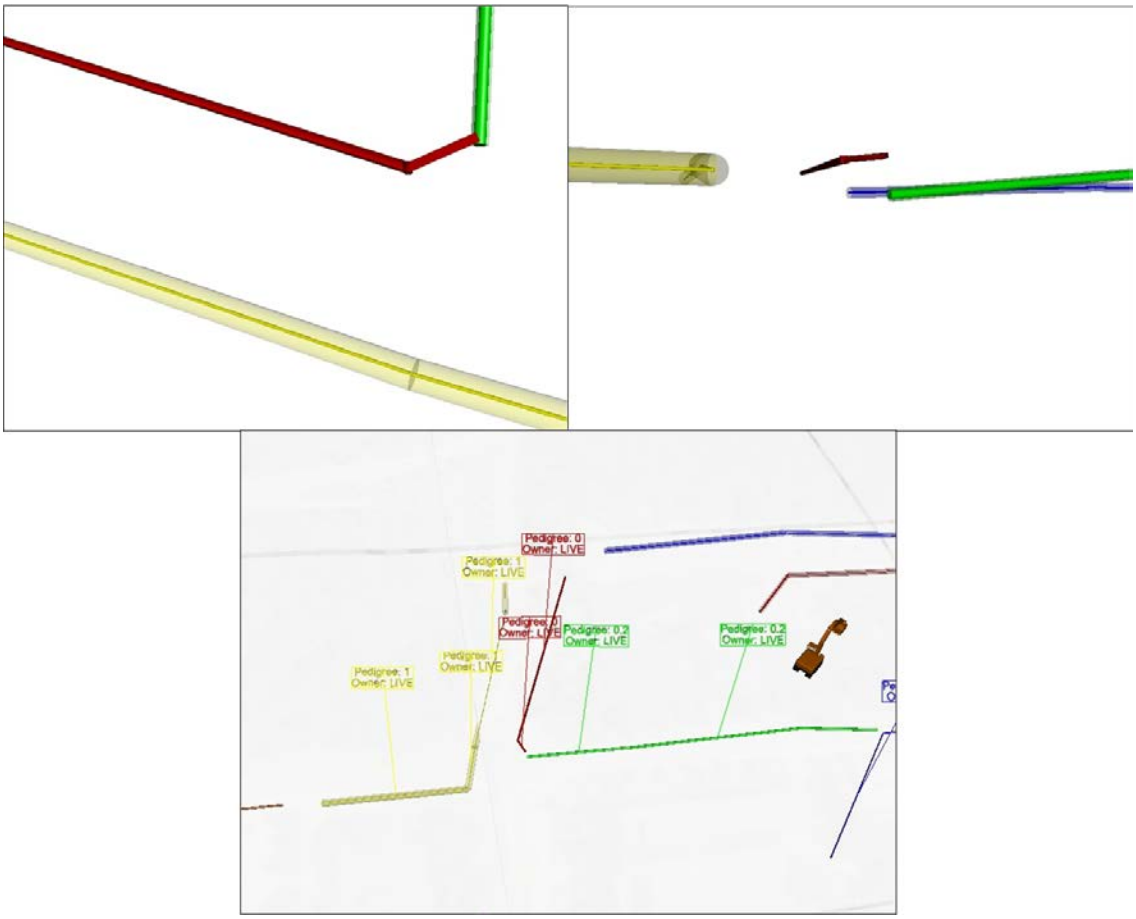

Fig. 3.15: Visualization showing terrain layer turned on/off and utility attributes shown at user's discretion

3.5.3 3D Modeling of Georeferenced Buried Utilities

The creation of 3D models of buried utilities that satisfy the requirements for IDEAL visualization implementation is one of the primary data sources. This module is called Buried Utility 3D Modeling Toolkit (B3M), and it enables the creation of georeferenced buried utility models. Like the visualization, it is also based on scene graph architecture, thus creating a direct link between the 3D models created and the visualization stage of the data. B3M is capable of creating models that satisfy the requirements of the IDEAL framework, for example the display of utility attributes and the representation of location uncertainty through a buffer or halo. The 3D utility models created by B3M adhere to the American Public Works Association (APWA) color-code, which the one-call marking systems also follow (APWA 2005). A polygon surface model was selected as the 3D model format, owing to its broad implementation and its ability to be used in downstream processes such as proximity monitoring.

B3M creates georeferenced 3D models that are location-aware. Thus the 3D utility models are registered to the same spatial referencing system as their surroundings. Figure 3.16 displays the locations of the utilities from their native GIS archival state, as well as the 3D visualization stage. The utilities can be seen having the same positions with respect to surrounding features, such as street intersections and lot boundaries. In addition, the 3D models also contain utility attribute information, as shown in Figure 3.17. The presence of attribute information in 3D models allows them to be seen in visualization, and can help utility inspectors and equipment operators make decisions that improve the productivity and effectiveness of planned engineering operations.

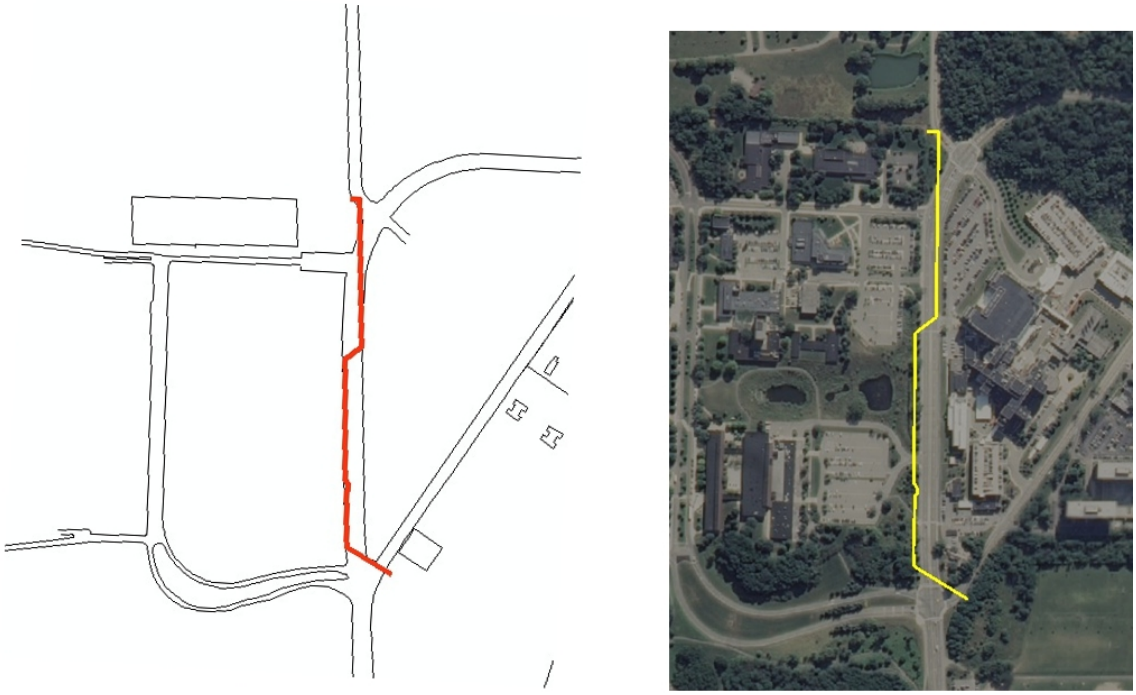

Fig. 3.16: Utility location comparison in GIS display (left) and 3D visualization framework (right) shows utility with respect to adjacent streets and pavement edges (Utility Data Source: DTE Energy, Detroit, Michigan)

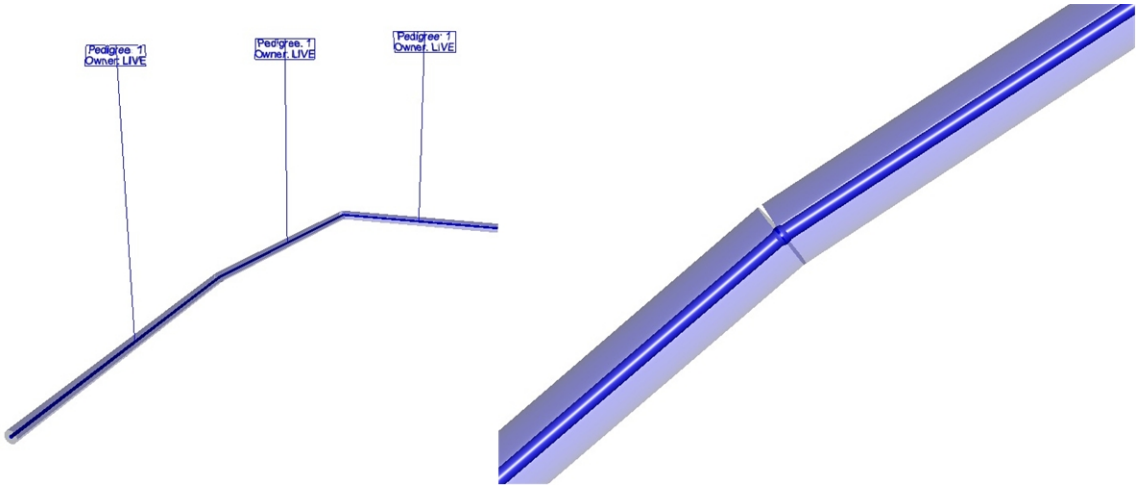

Fig. 3.17: Buried (water) utility model and attributes shown in a heads-up-display

As seen earlier in the Data Archival section, data sources for buried utilities can vary greatly from one organization to another, and the ability to handle multiple data input formats is a key attribute of B3M. The following section describes implementation details associated with buried utility data archival sources.

3.5.4 Sources of Input Geospatial Data

The most common formats used by organizations to store buried utility data are GIS, CAD, and XML. By ensuring that the 3D modeling framework can accept data in any of these formats, the visualization stage remains insulated from the data archival stage. The 3D buried utility models are output in OSG's native binary format, as this format can be used in any OSG-based visualization implementation. However, in addition to the native OSG format, output models can be created that are usable in other independent 3D visualization and modeling toolkits (OpenSceneGraph 2007b). The following sub-sections provide specific details regarding the most commonly used data archival formats.

3.5.4.1 Geospatial / Geographic Information Systems

In GIS, utility pipes are typically modeled as polyline features, and utility junctions as point features. Shape, size, depth, and other information are treated as utility-specific attributes. A single utility represented as a GIS polyline feature consists of a number of straight line segments connected at turning points but represented as a single entity. In the real world, a utility having bends consists of individual pipe segments that are joined to

one another to create an overall single entity. Thus a pre-processing step was included to break down GIS polylines into straight line features connected at turning locations prior to 3D modeling. Through this step, a 3D utility consists of individual 3D segments and not one single entity. Following the pre-processing step, the location, depth, cross-section shape, and dimension are retrieved from the GIS attributes to create 3D polygon models.

3.5.4.2 Computer-Aided Design (CAD) Systems

The B3M framework can create 3D utility models from CAD input data. Similar to the approach of treating GIS input, B3M creates a 3D polygon model for every straight line segment in a polyline. This arrangement ensures that every 3D model created is analogous to its representation in the real world. Unlike GIS, which treats spatial and non-spatial properties seamlessly, CAD requires its attribute data (e.g., shape, size, type, service dates, owner, etc.) to be archived in separate files that need to be accessed by the 3D modeling framework during creation of models.

3.5.4.3 Extensible Markup Language (XML) Formats

An XML document is referred to as an instance document. An instance document is based on an underlying structure that is referred to as a schema. An XML schema defines the elements and their order in an instance document, and is used to ensure that an instance document confirms to the preset structure. The data in instance documents is extracted through a process called parsing. Instance documents are checked for conformance with the schema document to ensure that data received by the modeling

framework is valid. As users have control over authoring schemas, all of the information required to create accurate and informative 3D models can be made available to the 3D modeling framework. Thus a schema document can be modified to reflect the data transmission method from the archival stage to the modeling and visualization stage.

3.5.5 Use of 3D Models in Geometric Proximity Detection

The IDEAL framework requires that a visualization be capable of extensible functionality beyond its core. The visualization implementation developed by the authors uses proximity monitoring and collision detection as the extensible feature. Geometric proximity monitoring is done through the implementation of the PQP library that uses polygon soup models as geometric input (PQP Gamma 1999). The input is provided in the form of a list of polygons or polygon soup that represent the surface of a 3D model. Utility models being georeferenced are location-aware, hence their polygons can be used directly in proximity monitoring analysis. Such visualization can assist excavation operators by providing real-time warnings and visual guidance to prevent accidental utility strikes.

The visualization's extensibility is tested by running a simulation of an excavator digging in close proximity to underground utilities. The authors would like to point out that the emphasis of this test is to investigate the ability of a visualization to provide guidance to excavation crews. Hence the position and orientation data that are used are simulated, and the testing environment is virtual. However, as the 3D models representing the utilities

and excavator are unaffected by the type of input tracking data, the determination of the suitability of this visualization for extensible proximity monitoring can be performed adequately through this 3D visual simulation. Figure 3.18 shows a virtual scene with an excavator, 3D terrain, and buried utilities where proximity monitoring is represented through a line joining the closest pair of points on the end-effector—buried utility pair. It also shows the use of an uncertainty buffer or halo around the utilities and end-effector to account for location errors.

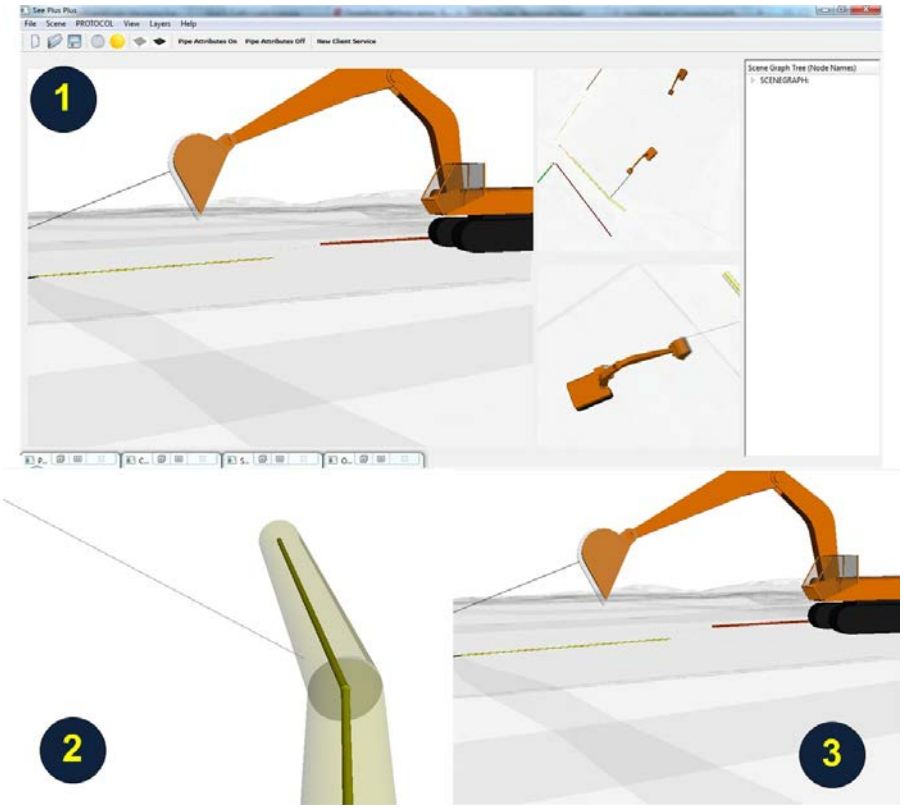

Fig. 3.18: 3D visualization of simulated excavation: (1) Overall user interface, (2) Zoomed-in view of buried utility, (3) Zoomed-in view of excavator

Proximity query tests are done in two iterations. In the first iteration, computations are made for distance between the excavator's end-effector and a buried utility for having zero uncertainty. In the second iteration, the buried utility has an uncertainty buffer of 1m, and the proximity query thus introduces an additional safety buffer to account for the uncertainty. Figure 3.19 shows screenshots of the visualization representing the two iterations carried out. The proximity monitoring information is displayed to the user in an easy-to-read heads-up-display in the scene. Together with real-time distance readouts, the presence of a wide halo or uncertainty buffer around a utility forewarns an operator about the quality (or lack thereof) for utility location data, and thus makes him/her aware of the need for additional caution.

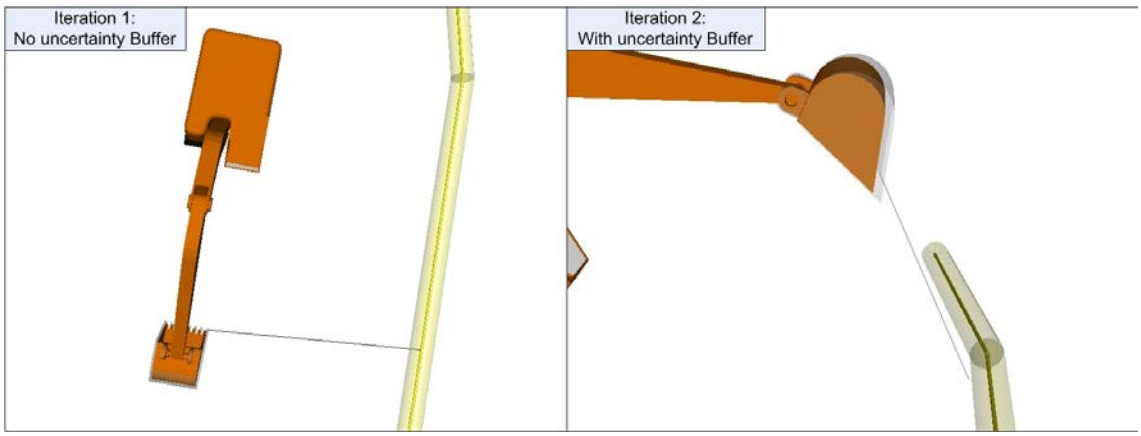

Fig. 3.19: (Left) Proximity between end-effector and utility ignoring uncertainty buffer and (Right) proximity accounting for uncertainty buffer in analysis

3.6 Conclusions and Future Work

This chapter described the details of the various stages in the life-cycle of underground utility geospatial data, and analyzed the inherent limitations that preclude the effective

use of the data in downstream engineering applications such as excavation guidance. Five key requirements—Interactivity, Information Richness, 3-Dimensionality, Accuracy Characterization, and Extensibility—were identified as necessary for the consumption of geospatial utility data in location-sensitive engineering applications. A visualization framework named IDEAL that meets the outlined requirements to guide the design and implementation of specific applications was developed and presented.

The framework was implemented through a real-time, user-interactive, 3D visualization. In addition, the creation of the georeferenced 3D models of buried utilities from XML data sources was described. The 3D models created are color-coded to the American Public Works Association standards, and contain attribute information and uncertainty buffers. Displaying the uncertainty associated with visualized utility location data is stated as being a key element in any visual guidance provided to excavator operators or field personnel to minimize accidental damage. Such uncertainty depends upon the accuracy of the tracking equipment used to collect the data, the skill of the field personnel doing the data collection, and the accuracy of the existing data that is used to create the dataset. The authors' future goal in this direction is to create improved uncertainty qualifying algorithms and present the information to excavator operators in practical field implementations of excavation guidance applications.

3.7 Acknowledgments

This presented research was partially funded by the US National Science Foundation (NSF) via Grants CMMI-825818 and CMMI-927475. The writers gratefully acknowledge NSF's support. The writers also thank the DTE Energy Company and ProStar Systems Inc. for their support in providing part of the utility data used in this research. Any opinions, findings, conclusions, and recommendations expressed in this chapter are those of the authors and do not necessarily reflect the views of the NSF, DTE, ProStar Systems, the University of Michigan, or Purdue University.

3.8 References

Anspach, J.H. (2011), "The Case for a National Utility As-Builting Standard", <http://content.asce.org/files/pdf/10Asbuiltstandard.pdf> (11/19/2011)

APWA (American Public Works Association) (2005), "Uniform Color Code", <http://www2.apwa.net/documents/About/TechSvcs/One-Call/COLORCC.PDF> (11/19/2011)

ASCE (American Society of Civil Engineers) (2002) "Standard Guidelines for the Collection and Depiction of Existing Subsurface Utility Data" ASCE Codes and Standards Activity Committee (CSAC), New York, N.Y.

ASCE (American Society of Civil Engineers) (2011), "Standard Guidelines for the Collection and Depiction of Existing Subsurface Utility Data", <http://www.fhwa.dot.gov/programadmin/asce.cfm> (11/19/2011)

Beck, A., Fu, G., Cohn, A.G., Bennett, B., Stell, J.G., (2007), "A framework for utility data integration in the UK", In: Proceedings of the 26th Urban Data Management Symposium, Stuttgart, Germany, 10–12 October.

Beck, A., Cohn, A.G., Parker, J., Boukhelifa, N., Fu, G. (2009) "Seeing the Unseen: delivering integrated underground utility data in the UK", Proceedings of the GeoWeb conference, Vancouver, July 2009.

Bernold, L.E. (2005a) "Accident Prevention Through Equipment Mounted Buried Utility Detection" Trends and Current Best Practices in Construction Safety and Health, Published 31st August, 2005, ISBN 1-886431-09-4.

Bernold, L.E. (2005b). "Automatic as-built generation with utility trenchers." Journal of Construction Engineering and Management, ASCE, 131(6), 740-747.

Bruce, C. (2011), “Location, location, location - GIS mapping of utilities is a growing business for some forward-thinking survey firms”, Point of Beginning, BNP Media, <http://www.pobonline.com/Articles/Features/BNP_GUID_9-5-2006_A_10000000000001073198> (11/19/2011)

CGA (2008), Common Ground Alliance - “Call before you dig”, <http://www.call811.com/for-the-media/once-per-minute.aspx> (10/05/2011)

Chen, J.Y.C.; Thropp, J.E. (2007) "Review of Low Frame Rate Effects on Human Performance," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on , vol.37, no.6, pp.1063-1076, Nov. 2007, doi: 10.1109/TSMCA.2007.904779.

Corbley, K. (2007), “Can you dig it?” The American City & County; Feb 2007; 122, 2; ABI/INFORM Global, pg. 32, <http://americancityandcounty.com/technology/government_dig/> (11/19/2011)

Cypas K., Parseliunas E., Aksamitauskas C. (2006), “Storage of underground utilities data in three-dimensional geoinformation system”, Geodetski vestnik. Vol. 50. No. 3. P. 481–491.

Evans, A.J., Roberts, G.W., Dodson, A.H., Cooper, S., Hollands, R., Denby, B., Turner, M., Owen, D. (2003) “Applications of augmented reality: Utility companies”, Survey Review 37: (289) p. 168-176, 2003.

FHWA (Federal Highway Administration) (1999), “Cost savings on highway projects utilizing subsurface utility engineering.” Final Rep.No. DTFH61-96-00090, Washington, D.C.

Grasmueck, M., Viggiano, D. A. (2007), "Integration of Ground-Penetrating Radar and Laser Position Sensors for Real-Time 3-D Data Fusion," *Geoscience and Remote Sensing, IEEE Transactions on* , vol.45, no.1, pp.130-137, Jan. 2007 doi: 10.1109/TGRS.2006.882253.

Griffin, J. (2011), “UIT, CAT partnership helps push technology envelope”, Underground Construction, Volume 66, Issue 1, Oildom Publishing Company of Texas, Inc.

Harper, I., Richardson, K. (2009), “A solid foundation for collaboration: GIS provides the backbone for accurate and seamless data collection for 100-kilometer pipeline project in New South Wales' Hunter Valley”, GEO: connexion, Bev-AL Communications, Inc, Volume 8, Issue 9.

Hochmuth, P. (2006), “Vegas water district mixes wireless, maps” Network World, Feb 27, 2006; 23, 8; ABI/INFORM Global, pg. 14”.

Huang, J., Cheng, B. (2009), "Interactive Visualization for 3D Pipelines Using Ajax3D," Networking and Digital Society, 2009. ICNDS '09. International Conference on , vol.1, no., pp.21-24, 30-31 May 2009 doi: 10.1109/ICNDS.2009.12.

Hubona, G.S., Shirah, G.W., Fout, D.G. (1997), “3D object recognition with motion”, *CHI '97* extended abstracts on Human factors in computing systems: looking to the future (CHI EA '97). ACM, New York, NY, USA, 345-346.

Jeong, H. S., C. A. Arboleda, D. M. Abraham, D. W. Halpin, and L. E. Bernold. (2003), “Imaging and Locating Buried Utilities”, Publication FHWA/IN/JTRP-2003/12, Joint Transportation Research Program, Indiana Department of Transportation and Purdue University, West Lafayette, Indiana, 2003. doi: 10.5703/1288284313237.

Junghanns, S., Schall, G., Schmalstieg, D. (2008) “Employing location-aware handheld augmented reality to assist utilities field personnel” Lecture Notes in Geoinformation and Cartography, Springer November 2008 <http://www.icg.tugraz.at/Members/schall/lbs_abstract.pdf> (06/20/2011)

Lew, J. J. (1996). "Subsurface Utility Engineering - An initial step in Project Development." The Proceedings of the Associated Schools of Construction, Texas A&M University, Texas, April 1996, 217-222.

Lin, M.C., Gottschalk, S. (1998), “Collision detection between geometric models: a survey”, <ftp://ftp.cs.unc.edu/pub/users/lin/cms98.pdf> (11/19/2011)

MapTools (2012), “Why Use UTM Coordinates”, <http://www.maptools.com/UsingUTM/whyUTM.html> (08/15/2012)

New York Times (1995), “Newark airport is closed as crew cuts power lines”, <http://www.nytimes.com/1995/01/10/nyregion/newark-airport-is-closed-as-crew-cuts-power-lines.html?pagewanted=2> (06/17/2011)

North, D. (2010), “Marking the Spot” CE News Archives, <http://www.cenews.com/print-magazinearticle-marking_the_spot-8115.html> (01/11/2012)

OpenSceneGraph (2007a), “Introduction”, About/Introduction <http://www.openscenegraph.org/projects/osg/wiki/About/Introduction> (06/26/2011)

OpenSceneGraph (2007b), “Plugins”, Support/User Guides/Plugins <http://www.openscenegraph.org/projects/osg/wiki/Support/UserGuides/Plugins> (11/19/2011)

Ordnance Survey (2004), “Getting closer with GPS”, News Release: Utilities using GPS – 09 December 2004 http://www.ordnancesurvey.co.uk/oswebsite/media/news/2004/dec/utilitiesgps.html> (11/19/2011)

Patel, A., Chasey, A. (2010), “Integrating GPS and laser technology to map underground utilities installed using open trench method”, Construction Research Congress 2010: Innovation for Reshaping Construction Practice [0-7844-1109-3].

PHMSA (2010), “Significant Pipeline Incidents through 2010 By Cause” <http://primis.phmsa.dot.gov/comm/reports/safety/SigPSIDet_2001_2010_US.html?nocache=9140#all> (10/31/2011)

PQP Gamma (1999), “PQP – A Proximity Query Package” <http://gamma.cs.unc.edu/SSV/> (6/22/2011)

Roberts, G.W., Hancock, C., Ogundipe, O., Meng, X., Taha, A., Montillet, J-P. (2007), “Positioning Buried Utilities Using an Integrated GNSS Approach”, In Proceedings of the International Global Navigation Satellite Systems Society (IGNSS) Symposium 2007, The University of New South Wales, Sydney, Australia, 4 – 6 December, 2007.

Schall G., Mendez E., Kruijff E., Veas E., Junghanns S., Reitinger B., Schmalstieg D. (2009), “Handheld Augmented Reality for Underground Infrastructure Visualization” In Personal and Ubiquitous Computing, Personal and Ubiquitous Computing. Vol. 13, no. 4, pp. 281-291. May 2009.

Sipes, L. (2007), “GIS for the Utilities Industry”, <http://www.cadalyst.com/gis/gis-utilities-industry-spatial-tech-column-9178> (11/22/2011)

Sterling, R. L. (2000), “Utility Locating Technologies: A Summary of responses to a statement of Need Distributed by the Federal Laboratory Consortium for Technology Transfer”, Trenchless Technology Center, Louisiana Tech University, Ruston, LA.

Stevens, R. E., and Anspach, J. H. (1993), “New technology overcomes the problems of underground system interferences on power projects.” Proceedings of the American Power Conference. Publ by Illinois Inst of Technology, Chicago, IL, USA. v 55 pt 1 1993. 323-326.

Talmaki, S. A., and Kamat, V. R. (2012), “Real-Time Hybrid Virtuality for Prevention of Excavation Related Utility Strikes”, Journal of Computing in Civil Engineering, American Society of Civil Engineers, Reston, VA. (In Review).

Tulloch, M., Hu, W. (2005), “A Proposed Solution For Mapping Underground Utilities For Buried Asset Management.” <http://www.ryerson.ca/content/dam/research/showcase/download/Mark_Tulloch_3rd_prize_2005.pdf> (11/13/2011)

Wilder, F. (2010), “The Fire Down Below”, Texas Observer, December 2, 2010 <http://www.texasobserver.org/cover-story/the-fire-down-below> (10/31/2011)

Wiss, U., Carr, D. (1998), “A Cognitive Classification Framework for 3-Dimensional Information Visualization”, Research Report LTU-TR---1998/4---SE, Luleå University of Technology. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.57.477&rep=rep1&type=pdf> (11/19/2011)

WRAL archives (2007), “Who's to Blame for Cary Gas Line Rupture?” <http://www.wral.com/news/local/story/1916911/> (06/17/2011)

Zlatanova, S., Stoter, J. (2006), “The role of DBMS in the new generation GIS architecture,” in: Rana&Sharma (Eds.) Frontiers of Geographic Information Technology, Springer-Verlag, Berlin Heidelberg ISBN-1- 3-540-25685-7, pp. 155-180, 2006.

Chapter 4

Active Geometric Proximity Monitoring of Visibility-Constrained Construction Processes Using Real-Time Graphical Simulation

4.1 Introduction

Civil Engineering projects are unique in nature as they lead to the creation of an unstructured, dynamic, and continuously evolving work space (Son et al. 2008). Projects in crowded urban areas present project participants with narrow, constrained work spaces and limited visibility of resources that increase the probability of collisions between equipment, workers, materials, and jobsite infrastructure (Teizer et al. 2010a, Cheng and Teizer 2011). Projects involving the use of large equipment increase the risk of collisions between equipment and workers, which is due to the reduced spatial awareness of operators as a result of blind spots on equipment and haul roads. Some projects require equipment to be operated remotely through a process called tele-operation in which operators control equipment with the aid of video cameras and wireless technology. Such operations introduce an additional obstacle in achieving spatial awareness due to the limited field of view afforded by on-board cameras (Chen et al. 2007).

The lack of clear visibility and spatial awareness result in accidents—such as workers-on-foot being struck by heavy equipment—and collisions between equipment and workers or between two pieces of equipment (Cheng and Teizer 2011). In addition to the above scenarios, certain projects inherently pose constraints on equipment operators' ability to

clearly perceive and analyze their working environment (Zhang et al. 2008). For example, excavator operators carrying out excavation operations in the presence of underground utilities are faced with the constant risk of striking buried utilities. Operators must rely on judgment and experience to determine the position of buried utilities in the absence of equipment and infrastructure tracking. Another example of the lack of clear visibility is the case of drilling equipment operators carrying out operations on reinforced concrete slabs. As the underlying reinforcement steel and utility conduits are hidden from the operator's view, the operator cannot be certain of correct drilling locations in the absence of drill-bit position-orientation tracking with respect to obstruction locations.

In 2009, visibility-related fatalities that included equipment accounted for a total of 521 fatalities due to workers being struck by moving equipment (Hinze and Teizer 2011). Excavation-related damage to buried utilities caused 544 major accidents, resulting in 37 fatalities, 152 injuries, and close to $200 million in property damage (PHMSA 2012). The high number of accidents resulting in injuries, fatalities, and monetary and productivity loss has thus led to significant research interest in tackling the problem of construction jobsite accidents that are related to reduced spatial awareness.

In this paper, the authors investigate the types of occurring spatial interactions and the need for real-time monitoring on construction jobsites. A computing framework is presented for monitoring interactions between mobile construction equipment and other jobsite entities, such as buried infrastructure, workers on foot, and other equipment. The framework is based on the use of sensor-based tracking and an evolving 3D graphical

database. Results from experiments conducted to analyze the achievable measurement error of the monitoring framework using indoor GPS tracking as the ground truth system are presented and discussed. The remainder of this paper is structured as follows: Section 4.2 provides an overview of equipment-related accidents on construction jobsites and currently adopted avoidance technologies and methodologies. Sections 4.3 and 4.4 present the limitations of current approaches and the authors' proposed methodology. Section 4.5 introduces the technical approach, and section 4.6 presents an implemented framework called PROTOCOL that is designed to provide the ability to perform real-time spatial queries in 3D visualizations of concurrent engineering processes. The results from validation experiments are provided in section 4.7. Limitations and future improvements are presented in section 4.8.

4.2 Literature Review

Due to the large number of accidents and fatalities on construction jobsites, significant prior research has been conducted to understand the root causes of these troubles. While the US construction industry accounts for only around 7% of the total workforce, it is responsible for nearly 20% of all industrial fatalities (MacCollum, 1995). The most common construction jobsite accidents are workers being struck by equipment or objects, workers falling into trenches or openings, electrocution, burns, and incidents involving the collapse of structures. Table 4.1 shows the number of fatalities caused due to workers being struck by an object or piece of equipment in the construction, mining, and manufacturing industries over a five-year period, as recorded by the Bureau of Labor Statistics (BLS) (BLS 2012).

Year	Construction	Mining	Manufacturing
2010	141	41	93
2009	154	34	95
2008	208	51	116
2007	215	71	138
2006	223	60	152

Table 4.1: Fatalities per year due to workers being struck by objects or equipment in the Construction, Mining, and Manufacturing sectors (BLS 2012)

It is estimated that approximately one-fourth of construction worker deaths are the result of collisions, rollovers, struck-by accidents, and a variety of other equipment-related incidents (Hinze and Teizer 2011). One of the chief contributing factors is workers-on-foot and large mobile equipment sharing the same workspace. Appreciation of the risks involved has resulted in rules and policies governing the operation of equipment on construction jobsites. Occupational Safety and Health Administration (OSHA) regulations 29 CFR 1926 Subpart O, 1926.601(b)(4) and 1926.602(a)(9) require mobile construction equipment to be equipped with operational back-up alarms and/or a spotter that signals when the equipment can safely back up (OSHA 2010). There is a requirement for the alarms to be audible even in the presence of the high ambient noise that is typical on a construction jobsite (OSHA 2010). These regulations further reinforce the fact that equipment operators suffer from reduced visibility and spatial awareness.

The situation is more severe during nighttime construction due to the lack of natural light and poorly lit workspaces. A large percentage of highway construction takes place at night in order to reduce its adverse effects on traffic flow (Arditi et al. 2005). Thus the lack of visibility becomes a greater contributing factor. Poor lighting and visibility-related issues were found to be one of the leading causes of accidents involving through traffic and workers, and construction equipment and workers (Arditi et al. 2005).

The following statistics on worker safety from the Federal Highway Administration (FHWA) and BLS show that lack of clear visibility and spatial awareness is a major cause of a large number of accidents. Nearly half of worker fatalities are caused when workers are run over or backed over by vehicles or mobile equipment. Between 2005 and 2010, runovers/backovers were the cause of an average of 48% of worker fatalities. The second most common cause of worker fatalities was collisions between vehicles/mobile equipment. The third most common cause of worker fatalities was workers caught between or struck by construction equipment and objects, which between 2005 and 2010, this was the cause of an average of 14% of worker fatalities. (FHWA 2012).

Operations that involve concealed or buried infrastructure—such as excavation, trenching, and drilling—also pose visibility and spatial constraints on operators. The lack of knowledge of the equipment end-effectors' location in relation to concealed infrastructure results in inadvertent strikes. Thus, there is a clear need to supplement workers' and equipment operators' visibility through additional means, thus providing increased spatial awareness. Prior research addressing these issues can generally be

classified into: 1) Vision-based environment recreation approaches and 2) Position-Orientation tracking-based approaches, as summarized in Table 4.2.

Vision-based Jobsite Reconstruction	Position-Orientation Tracking Sensor
3D Imaging Sensors (Example - Laser scanning - sparse point clouds, dense point clouds, and Flash Laser Detection and Ranging i.e. LADAR)	Global Positioning System (GPS)
	Radio Frequency Identification (RFID)
	Radio Frequency (RF) sensing and actuating
Computer Vision – object recognition	Ultra-wide band (UWB)

Table 4.2: Existing implementations and technologies can be divided into sensor-based approaches and vision-based approaches

Several technologies have been used in order to improve positional awareness and knowledge of the surroundings. A preliminary real-time crash avoidance framework using 3D imaging sensors for equipment control was presented by Chi et al. (2008). Kim et al. (2006) demonstrated the use of a laser rangefinder, along with pan and tilt kinematics, to recreate the jobsite using convex hulls and workspace partitioning. Simulations involving Obstacle Avoidance, Artificial-Potential Function, and Minimum Distance Algorithms showed the potential of a jobsite warning system in preventing collisions (Kim et al. 2006). Sparse range point clouds using laser range finders have also been used to produce a rapid 3D modeling approach with less than 5% error for object recognition (Kwon et al. 2004). Cho et al. (2002) presented a new method for rapid modeling and visualization of a local area based on geometric information about objects

obtained using simple sensors (such as a single-axis laser rangefinder and a video camera) for better planning and control of construction equipment operations in unstructured workspaces. Son et al. (2008, 2010) presented the use of flash LADARs for real-time 3D modeling of a construction worksite for autonomous heavy equipment operation.

Radio Frequency (RF) technology has been implemented to provide real-time warnings to RF-tagged equipment operators and workers by triggering alarms when the distance between equipment and worker drops below a safety threshold (Teizer et al. 2010b). Low-frequency, low-power magnetic fields-based Radio Frequqncy Identification (RFID) technology was employed by Schiffbauer and Mowrey (2001) to provide warnings when workers and heavy equipment came too close to each other. Oloufa et al. (2002) have demonstrated the use of Global Positioning System (GPS)-based tracking for providing warnings about impending collisions between equipment by calculating the intersection point of the two vectors representing two moving vehicles. We refer the reader to Zhang et al. (2008) for a comprehensive and informative survey and references to other work using GPS for equipment tracking and automated control of construction equipment. Hwang (2011) also demonstrated the use of Ultra-wide band (UWB) technology to monitor the positions of crane booms, and to provide warnings when a pair of cranes comes within a safety threshold distance of each other. Ruff and Hession-Kunz (2001) implemented Radio Frequency Identification (RFID) technology in a prototype system to provide warnings for a front-end loader to prevent collisions with workers-on-foot.

4.3 Limitations of Existing Methodologies

Any collision detection or proximity monitoring algorithm between equipment, personnel, and/or jobsite infrastructure involves at least two entities. The entities can be static, such as on-site materials and jobsite infrastructure, or dynamic, such as equipment and workers-on-foot. It is required that a safety monitoring system on a busy jobsite be capable of tracking distance on a one-to-many basis. Table 4.3 summarizes the possible interactions between various entities present on a typical construction jobsite.

Entity I	Entity II
Equipment	Equipment
Equipment	Worker-on-foot
Equipment	Jobsite infrastructure
Equipment	Jobsite materials
Worker-on-foot	Jobsite infrastructure
Worker-on-foot	Jobsite materials
Jobsite materials	Jobsite infrastructure

Table 4.3: Types of interactions between entities on a construction jobsite (Assumes that a piece of equipment and its operator are viewed as a single entity)

The existing collision detection or proximity monitoring methodologies can be divided into two groups. The first group consists of those technologies in which tracking sensors can be placed on entities 1 and 2 from Table 4.3. Technologies such as GPS, RF-transmission, UWB, and RFID, all require sensors to be present on both entities in order

to obtain their location for use in proximity analyses. The second group of technologies includes those that re-create jobsite infrastructure and layout in 3D models for use in real-time analysis. Thus there is a requirement for entities such as materials and infrastructure to be in the line-of-sight of these sensing technologies (such as 3D imaging sensors).

The unique and diverse nature of construction jobsites means that no single technology can be applied to all scenarios owing to physical and practical constraints. For example, underwater construction will pose limitations on the use of GPS and RF transmission for position tracking. Similarly, operations where infrastructure is covered by soil and/or concrete will limit the efficacy of vision-based sensing technologies. Object modeling and/or recognition requires that the object be visible to laser scanners and cameras. This is not possible for rebar embedded in concrete and buried utilities.

Operations such as these require the knowledge of the dynamic entity's (e.g., excavator, drilling equipment) location, as well as the embedded entity's location. Tracking the former entity is possible through the use of tracking sensors such as GPS. However, the latter entity involved in the query is neither capable of being instrumented with sensors nor visible to vision-based technologies such as laser scanners. As a result, the existing methodologies cannot be directly applied to such operations. A new methodology combining tracking sensors instrumented on the dynamic entity (entity I) and a graphical database representing the static entity (entity II) is thus developed and proposed to assist equipment operators in gathering the necessary spatial awareness. The following section

describes the proposed methodology using construction operations involving underground or out-of-sight infrastructure as a motivating example.

4.4 Proposed Methodology

Operations such as excavation in the presence of buried utilities pose additional challenges to equipment operators. Lack of accurate knowledge about the location of the hidden infrastructure in elevation (vertical plane) and plan (horizontal plane) reduces the spatial awareness of an operator with respect to their work zone. For example, excavating beyond the permissible safe depth in the presence of buried utilities can trigger accidents, leading to loss of life and damage to equipment and property (PHMSA 2012). In order to address this situation, a methodology involving the combination of sensor-based tracking and concurrent analysis of a graphical database is proposed. While the developed methodology can be applied to any operation involving hidden infrastructure, the example of buried utility excavation is used herein for illustration purposes.

An excavator operator digging in the presence of subsurface utilities (as seen in Figure 4.1) is primarily concerned with inadvertent contact with underlying pipes, cables, and conduits. To be able to dig safely and efficiently, an operator must know the location of the excavator's digging implement relative to buried utilities lying in close proximity. An effective means to convey this information to the operator is visually. Furthermore, the presence of audio-visual warnings to inform the operator of impending collisions between digging implements and underlying utilities can help prevent potential accidents.

Fig. 4.1: Limited visibility facing excavator operators and uncertainty regarding location of buried utilities

The basic proximity monitoring relationship is between an excavator (specifically its digging implement) and a single buried utility. Warnings and real-time 3D visualization are achieved by: 1) instrumenting the dynamic entity in the relationship with tracking sensors (for example: instrumenting an excavator with position and tilt sensors); 2) creating a 3D graphical database to represent the static and dynamic entities; and 3) analyzing interactions between entities in a real-time 3D visualization. Figure 4.2 illustrates the proposed methodology. The dynamic entity's 3D model is updated in real-time through tracking sensors at the jobsite. Thus the 3D visualization serves as a simulation of the real-world operation. As the operation proceeds, the graphical database is analyzed for proximity as well as collisions between included entities. Results from the analysis can be used to trigger audio-visual warnings to notify the operator with vital information such as: 1) proximity of excavator's digging implement to underlying utility;

2) breach of safety threshold; and 3) impending collision. The post-analysis reaction is tailored to suit the requirements of a specific operation.

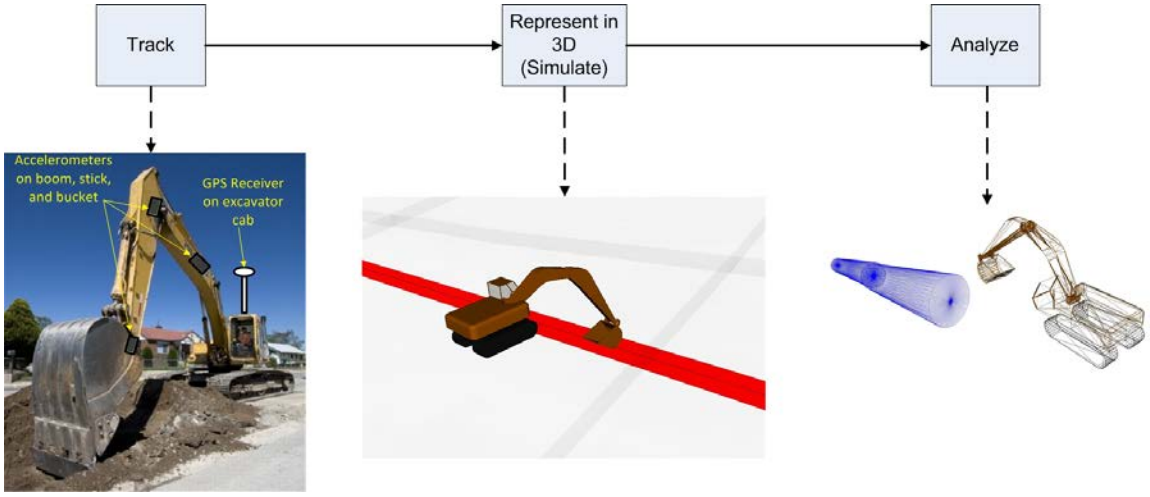

Fig. 4.2: Proposed Methodology: Track equipment on jobsite, represent in real-time 3D simulation, analyze 3D graphical database for proximity and collisions

The ability to view operations in real time improves the overall awareness of operators and site managers. Research indicates that construction operations—such as earth moving, heavy lifting, material handling, and remote excavation in cofferdams—can be performed more safely and effectively by using graphical models of both the equipment and the workspace (Kwon et al. 2004; Cho et al. 2002; Huang and Bernold 1997). Thus, in addition to audio-visual warnings for the operator, a real-time 3D visualization of the operation is also an essential component of the proposed methodology.

4.5 Technical Approach

The previous section introduced the proposed methodology to enable tracking for operations involving buried infrastructure. The methodology, as shown in Figure 4.2, consists of three stages: Track, Represent in 3D, and Analyze. The technical approach is applied to an operation involving a dynamic entity, such as an excavator, and one or many underground utility lines. However, the same approach can be applied to any operation involving a pair of dynamic entities or static entities, such as material or jobsite infrastructure, as given in Table 4.3. This approach requires real-time position-orientation tracking of dynamic entities and 3D geometric models representing the dimensions and positions of static entities. The final stage is completed by analyzing the geometric database consisting of all geometric entities involved in the operation.

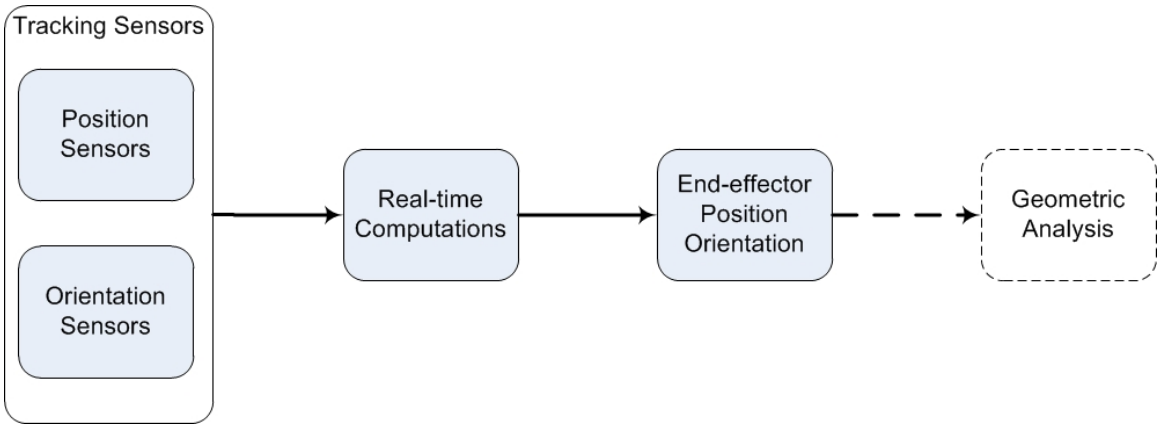

Fig. 4.3: End-effector position-orientation computation through sensor fusion and real-time computation

Efficiency, accuracy, interactivity, ability to handle dynamic objects, and capability to process a large number of 3D CAD model shapes, sizes, and forms have been identified

as essential technical requirements of any method designed for collision detection and interference analysis of 3D virtual construction simulations (Kamat and Martinez 2007). Also imperative is the ability to perform proximity queries in real time by processing position and orientation data. Output from the analysis stage—such as proximity between entities, potential collisions, and breaching of safety thresholds—can be used by downstream processes such as operator warning frameworks and equipment control mechanisms. The following sections describe the technical details of individual components of the proposed approach.

4.5.1 Track

Excavation operations typically involve infrastructure that is covered by earth, mud, soil, and debris. Equipment is used to remove the overlying cover. Excavation equipment has end-effectors (e.g., buckets) that interact with the materials overlying the buried infrastructure. Operators of such equipment cannot be certain of the exact location of the end-effector in relation to buried infrastructure. In order for such operations to be carried out in a safe manner, accurate knowledge of the pose (position and orientation) of the equipment's end-effector is critical. Knowledge of the end-effector's pose can then be used in the analysis stage to compute spatial relationships with infrastructure in its immediate surroundings. Hence, the primary step in carrying out geometric computations between jobsite entities is tracking the poses of equipment and, in particular, their end-effectors.

4.5.1.1 End-effector position and orientation

Due to the harsh nature of the work environment and operation, sensors cannot typically be placed directly on the digging implement of an excavator. In order to arrive at the pose of the end-effector, a combination of multiple sensors and real-time computations is employed. An excavator's end-effector is its bucket (bucket teeth). However, it is not practical to place GPS sensors in the bucket owing to potential damage due to constant interaction with the soil. The end-effector's pose is obtained through the use of a GPS sensor placed on the excavator cab, and orientation- or tilt-measuring sensors placed along the rotating joints for the boom, stick, and bucket, as described in Talmaki and Kamat (2012). The use of an indoor GPS tracking system has been demonstrated to calibrate the sensors on board a backhoe loader and track its position and orientations while performing operations in real time (Kini et al. 2009).

As the pose of the end-effector cannot be directly obtained, computations are made with output from multiple position and orientation sensors placed on the equipment, as schematically represented in Figure 4.3. Real-time calculations are performed to derive the end-effector pose. Thus the tracking stage is responsible for determining the pose of the dynamic entity's end-effector or component that interacts with other jobsite entities. Pose determination is done through a process of forward kinematics. In forward kinematics, the final shape of the articulated chain is dependent on the angles explicitly specified for each of its links (Kamat and Martinez 2005). The output has two components to it: Position and Orientation. A six-parameter vector is used to describe the spatial location and orientation of an object in space. A similar approach has been

successfully implemented in autonomous navigation and aircraft auto-pilot systems (Dasgupta and Banerjee 2006). The six parameters making up the vector are Latitude, Longitude, Altitude, Roll, Pitch, and Yaw.

Position sensors such as GPS compute locations in terms of geodetic latitude, longitude, and altitude above an imaginary ellipsoid. These locations are provided with respect to a specific geodetic datum or reference coordinate system. For GPS, the datum is WGS-84 (Dana 2000). Position coordinates are sometimes converted from spherical units, such as latitude-longitude, to planar systems, such as Universal Transverse Marcator (UTM), where the latter are more accurate at representing smaller areas by preserving angles and not distorting shapes. Planar systems employ linear units, such as meter and foot, and thus measurements made in them can be used directly in distance and other computations, unlike spherical units. (Geokov 2012).

Orientation is defined in terms of a triplet of parameters that are traditionally used to illustrate the rotation of an aircraft in aeronautical domains (Preston et al. 2005). The triplet of roll, pitch, and yaw compute the rotation of a body about its longitudinal, transverse (lateral), and vertical axes, respectively. Thus the output from the ‘Track’ stage is used to update dynamic entities and analyze their interactions with other jobsite entities, and also to create a graphical simulation of the jobsite.

4.5.2 Representation in 3D

The second stage of the proposed technical approach is representing the jobsite operation being performed in a real-time 3D visualization. The use of telepresence and virtual reality has been shown to have promise for the visualization and study of hostile and extreme environments (Hine et al. 1994). Virtual reality allows users to see beyond what is possible through conventional video camera feeds. The authors envision that a virtual reality scene representing a city block can be manipulated by a utility inspector to alter the transparency of the surface of terrain, and to show buried infrastructure in that area. 3D virtual scenes also enable operators to enjoy views of their equipment and surroundings that are not feasible through the placement of conventional cameras with a much narrower field of view (Huber et al. 2009).

4.5.2.1 Types of 3D Models

3D models or geometry can be represented in a variety of formats, such as wireframe models, surface models, solid models, meshes, and polygon soups. Depending on the level of realism required, 3D models can vary in accuracy, memory requirements, difficulty of modeling, and use in real-time rendering and analysis. However the choice of 3D geometry format also depends on the end application and real-world entity being modeled. Polygonal meshes are the most commonly used type of format to represent 3D geometry. Their popularity is also enhanced due to their versatility and ease of rendering (Lin and Gottschalk 1998, McHenry and Bajcsy 2008, Shen et al. 2004).

Parametric representations of surfaces typically define a surface as a set of points, and are commonly used in many commercial modeling systems (Bajaj et al. 1996). Non-Uniform Rational B-Splines (or NURBS) is an example of parametric methods. Applications requiring the use of a truly smooth surface at every scale find NURBS to be a convenient option to implement (McHenry and Bajcsy 2008). Implicit surfaces define a set of points that satisfy a function F where $F(x, y, z) = 0$, and all points that satisfy the criteria $F(x, y, z) < 0$ define a solid that is bounded by the implicit surface. Constructive Solid Geometry (CSG) allows 3D shapes to be built from simpler primitives—such as cubes, cylinders, and spheres—through the application of boolean operations on them. 3D file formats used within the CAD domain often store their geometric data such that CSG principles can be used to operate upon them (McHenry and Bajcsy 2008).

Boundary Representation (or B-Rep) describes a solid in terms of its surface boundaries (i.e., vertices, edges, and faces) (Durikovic 2012). In B-rep, a data structure containing information about an object's edges, faces, vertices, and their topological relationships is used to represent it. Figure 4.4 shows the classification of 3D Models into the above-described formats. For a detailed review of 3D model types, the reader is referred to Foley et al. (1992) and Hearn and Baker (1994). Polygon models are the most suitable format for hardware-accelerated rendering, and are the most commonly used format for representing 3D models (Lin and Gottschalk 1998, McHenry and Bajcsy 2008, Shen et al. 2004). Hence the polygon soup class of polygon surface models was selected as the 3D model format to represent real-world entities and perform geometric analysis.

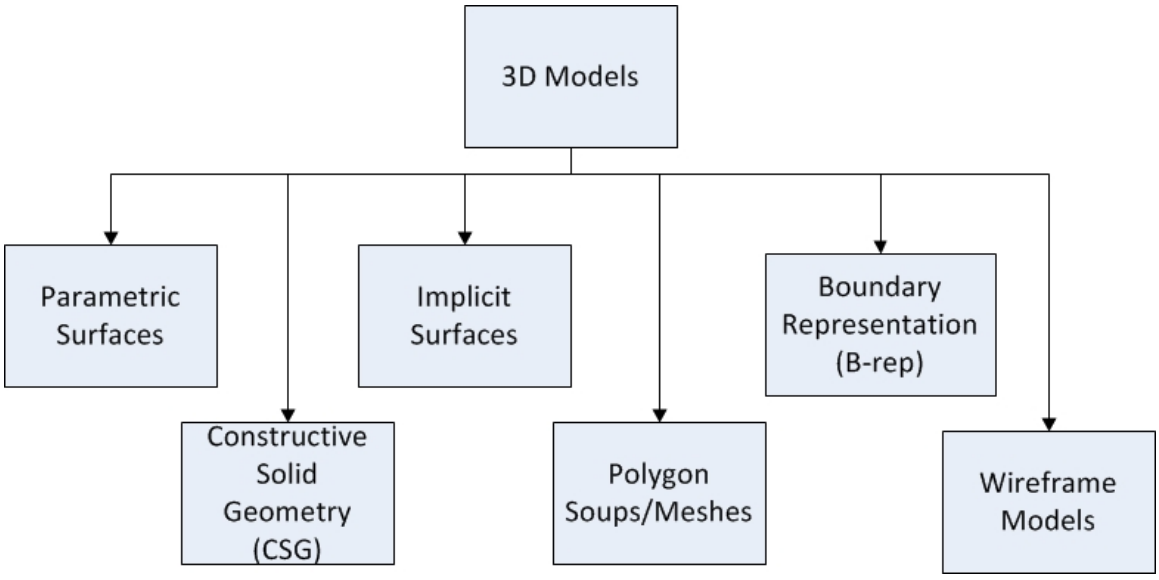

Fig. 4.4: Classification of 3D Model Representations

4.5.2.2 Georeferenced 3D Models – Static Entities

Dynamic entities on a jobsite are tracked using position and orientation sensors. Their equivalent 3D models in a real-time simulation are updated through position and orientation output from the 'Track' stage of the framework. However, in order to monitor spatial relationships between static and dynamic entities, equivalent positional information must be made available regarding the location of static entities. This is made possible through the use of georeferenced or location-aware 3D models. Georeferencing refers to the ability to assign a location attribute to data. The location attribute can be obtained through archived data sources such as Geographic Information System (GIS) databases, infrastructure repositories, and as-built drawings. Static entities such as underground utilities, embedded rebar, temporary structures, and material stockpiles have their location information linked to the earth's surface.

Thus, georeferencing ensures that 3D models representing static entities can be used in the same analysis with dynamic entities that utilize position-orientation information having a common coordinate system. Specifying the location of an object on the earth's surface is done through coordinate systems, such as Geographic Coordinate Systems (GCS) that use spherical units (latitude-longitude), or Projection Coordinate Systems (PCS) that use linear units. When creating a 3D simulation, it is important to note that all position-orientation data and georeferenced 3D models share the same coordinate system and units. Dissimilar systems and/or units can result in 3D models appearing farther away than in reality, and the resulting analysis will also produce unusable output. Thus, spatial analysis between dynamic and static entities is made possible through the use of georeferenced 3D models.

4.5.2.3 3D Model updates – Dynamic Entities

In order for a 3D visualization to simulate an ongoing operation, all dynamic entities must update their pose to match their real-world counterparts. In addition to the 3D visualization stage, updates made to dynamic entities are used downstream in the 'Analysis' stage. 3D Models representing dynamic entities thus afford equipment operators a virtual 3D view of the operation they are performing, and concurrent geometric proximity analyses between jobsite entities.

4.5.3 Analyze

The analysis of interference and/or interaction between geometric entities is intended to help equipment operators by providing improved spatial awareness. This can only be

achieved if all computations are carried out in real time (i.e. concurrently with the real-world operation). It thus follows that the efficiency of the algorithms implemented in the analysis stage is of critical importance. At the same time, accuracy of the chosen algorithm cannot be sacrificed in order to achieve optimum real-time or near real-time performance.

4.5.3.1 Creation of 3D graphical database

A class of polygon surface models known as polygon soups are made up of several hundred geometric primitives (e.g., triangles and quads), and are often present in an un-orderly arrangement with no topological relationships between adjacent polygons. Figure 4.5 shows the 3D model of a dump truck and its corresponding polygon surface model, showing the composition of its surface. Such a collection of polygons is commonly referred to as a polygon soup. When a proximity query is made between a pair of entities, the objects' geometric primitives and their global positions and orientations combine to create the graphical database for the query at hand, as shown in Figure 4.6.

Computational queries provide valid output as long as their underlying graphical database is current in terms of geometric makeup and object pose. The geometric primitive component of the graphical database remains the same unless there is a change in equipment type or configuration (e.g., a change of end-effector for an excavator). The pose is updated in real time and maintains the fidelity of the database. Thus, it is important to point out that both geometric content and pose information must represent their real-world counterparts at all times to ensure meaningful analysis. Figure 4.6 shows

the breakdown of 3D models into their geometric primitives for illustration purposes only, as the analysis of the graphical database is performed in a multi-stage manner and involves operations at the object and sub-object levels, but not always at the primitive level, as seen in the following section.

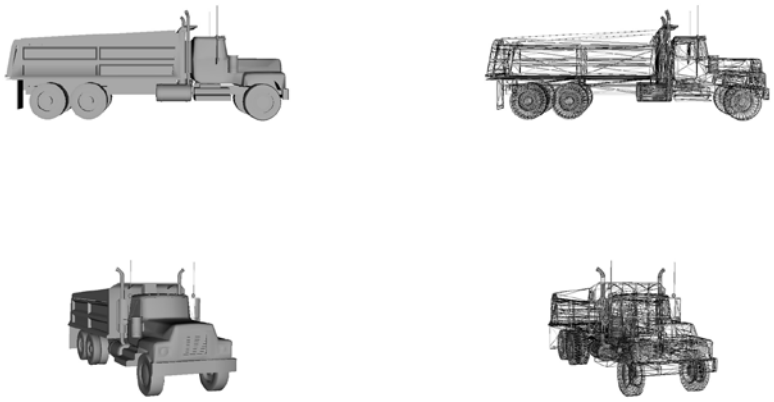

Fig. 4.5: Polygon surface models and their corresponding polygonal/wireframe view

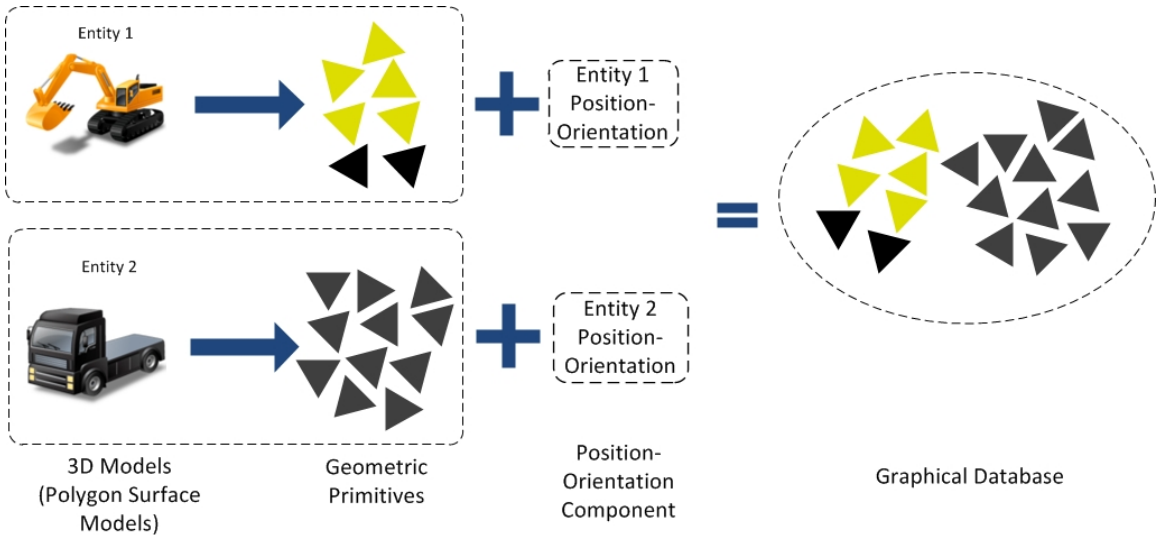

Fig. 4.6: Creation of graphical database from 3D models' polygon soups and their global position-orientation

4.5.3.2 Analysis of 3D Graphical Database

Algorithms for the detection of interference between objects are an essential component in virtual environments. Their applications are widespread and can be found in areas ranging from surgery simulation and games to cloth simulation and virtual prototyping (Teschner et al. 2005). The set of spatial queries between a pair of objects are collision detection, exact separation distance computation, and approximate distance measurement to a tolerance value (Larsen et al. 1999).

A bounding volume (BV) is used to bound or contain sets of geometric primitives such as triangles, polygons, and NURBS (Larsen et al. 1999). The most common types of Bounding Volumes (BVs) are spheres, axis-aligned bounding boxes (AABBs), oriented bounding boxes (OBBs), discrete oriented polytopes (k-DOPs), ellipsoids, convex hulls, and sphere swept volumes (SSVs) (Larsen et al. 1999, Teschner et al. 2005). The efficiency of a bounding volume is affected by the choice of BV type. Efficiency is achieved by a trade-off between the tightness of fit and the speed of operations between two such BVs. No single BV is optimal for all situations and queries (Larsen et al. 1999).

The underlying algorithm in the developed methodology uses sphere swept volumes as the BV type. First- and second-order statistical methods are used to compute the BVs. The mean and covariance matrix are computed for vertices of all triangles making up the object, and are used to summarize the object's primitive vertex coordinates. Computation details and formulae are described in detail in Gottschalk et al. (1996) and Larsen et al.

(1999). An OBB is first constructed to enclose the underlying geometry using the Eigen vector of the covariance matrix computed for vertex coordinates. The second step consists of creating BVs of SSV type.

There are three types of SSV: Point, Line, and Rectangle SSV. The selection of SSV type is based on the dimensions of the OBB constructed. The BVs are arranged in a hierarchical manner beginning with the root node that encompasses all geometry associated with an object, and ending with leaf nodes that contain only a single triangle. The BVs thus created at different node levels form a Bounding Volume Hierarchy (BVH) using a top-down strategy. All of the triangles in a given node are split into two subsets, where each subset then becomes a child node of the current node. This subdivision continues until a given node contains only a single triangle or primitive and can no longer be further sub-divided.

Triangles in a node are subdivided based on a splitting rule described in Gottschalk et al. (1996). The first step of the subdivision process is the selection of a suitable axis. The subdivision rule that is adopted in this procedure uses the longest axis of a box (i.e., OBB created for the geometry). If the longest axis cannot be selected then the second longest axis is chosen, or else the shortest axis is chosen. In the next step, a plane orthogonal to the selected axis is constructed. This plane acts as the partitioning plane such that polygons are divided into two groups according to which side of the plane their center point lies on. The subdivision process continues until a group of polygons cannot be

further partitioned by this criterion (i.e., the group is considered indivisible or it encounters a leaf node that contains only a single triangle/primitive).

Traversal of a BVH during a proximity query is referred to as bounding volume tree traversal (BVTT). The traversal begins at the root node and proceeds along the tree in a depth-first or breadth-first manner. The distance query returns the exact distance between a pair of objects, while the collision query returns whether or not they are overlapping, as illustrated in Figure 4.7.

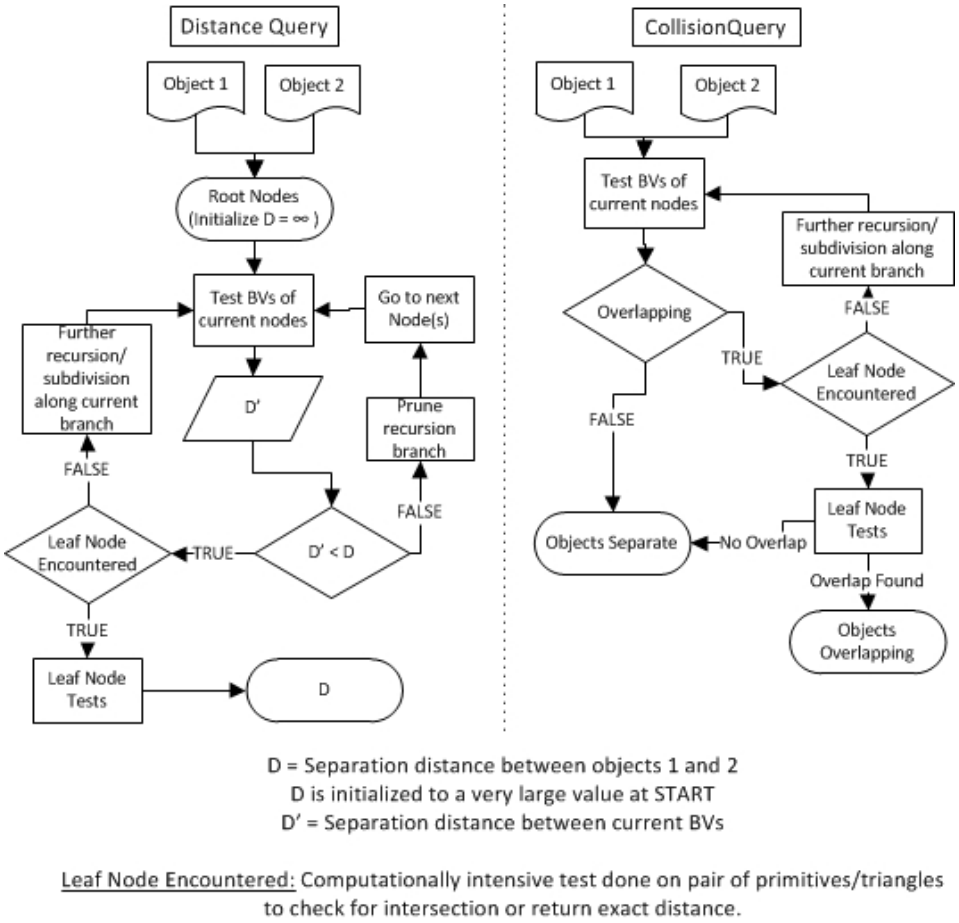

Fig. 4.7: Flow chart illustrating the progression of distance and collision queries in underlying algorithms

Overlap and distance computations are performed using a multi-stage approach. The first stage is less computationally intensive and is used to eliminate a large number of test pairs from being passed on to the more computationally demanding second stage. In the first stage, OBBs are used to determine if two objects are disjoint or overlapping. The same computation is used by overlap as well as distance tests to eliminate or prune object pairs from further checks. The algorithm is based on the separating axis theorem that is described in detail in Gottschalk et al. (1996). A pair of bounding boxes to be checked for overlap is projected onto a random axis in 3D space. In this axial projection, each box forms an 'interval' on the random axis. An interval is the distance between two projected points on a random axis. If the intervals of the boxes do not overlap, then the pair is disjoint/separate for that given axis. Such an axis is termed a 'separating axis'. The separating axis procedure is graphically represented in Figure 4.8. Once an overlapping axis is found, no further tests are required to check for overlapping.

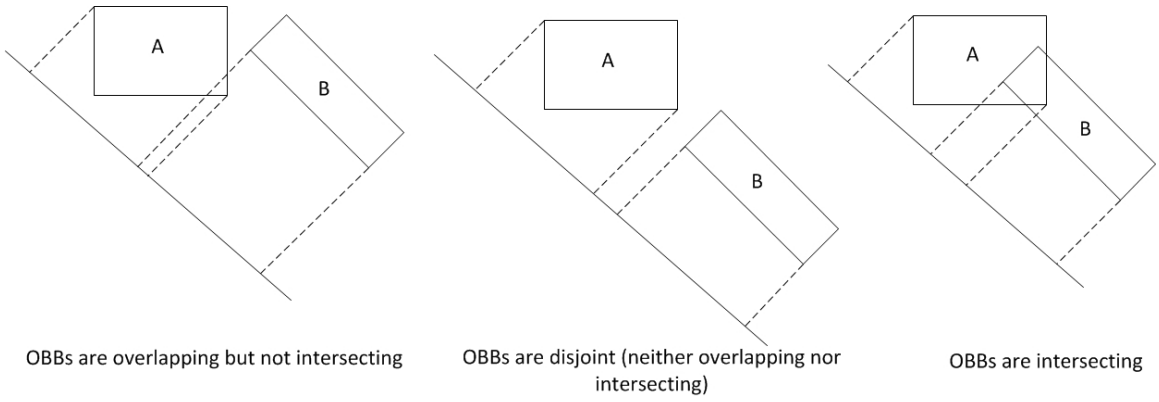

Fig. 4.8: Use of separating axis to determine overlap between OBBs

If, on the other hand, the intervals are found to be overlapping, then further tests are required to concretely determine whether the pair is separate or overlapping. Two boxes are disjointed if and only if a separation axis exists that is orthogonal to a face of either box or orthogonal to an edge from each box. Each box has three unique faces and three exclusive edge orientations. Thus a total of fifteen (six face combinations and nine pair-wise edge combinations) separating axes exist for a given pair of boxes. If no separating axis exists, then the boxes are overlapping; and if the current OBB is a leaf node (i.e., bottom-most node enclosing a geometric primitive), then the pair is passed to stage two for further computations.

The second stage is more computationally demanding and checks for overlap between underlying geometric primitives. If primitives are found to intersect (i.e., no separation distance between primitives), the computation—which uses a list of primitives making up the object—determines which primitives on each object are intersecting, and their exact points of intersection. If primitives are found to be separate, the separation distance between them is calculated. Computation and algorithm details involved in the stage two leaf-node tests are described in greater detail in Larsen et al. (1999). The algorithm first determines if the closest points between the primitives lie on their boundary edges. This is determined through the use of external Voronoi regions. An external Voronoi region for an edge A of a primitive is defined as the region of space outside the primitive in which all points are closer to edge A than any other features of this primitive.

According to the lemma used in the algorithm in Larsen et al. (1999), there are three possible configurations for a pair of edges, A and B, belonging to two primitives: (1) Edge B is present entirely inside the Voronoi of A; (2) B is present entirely outside the Voronoi of A; and (3) some points of B are inside the Voronoi of A, and some are outside, as shown in Figure 4.9. From a possible sixteen pairs of edges, if no pair satisfies the requirements of the lemma, then primitives are either overlapping or the closest point lies in their interior. Cases one and two are simple acceptance rejection configurations, while case three requires additional tests to be performed.

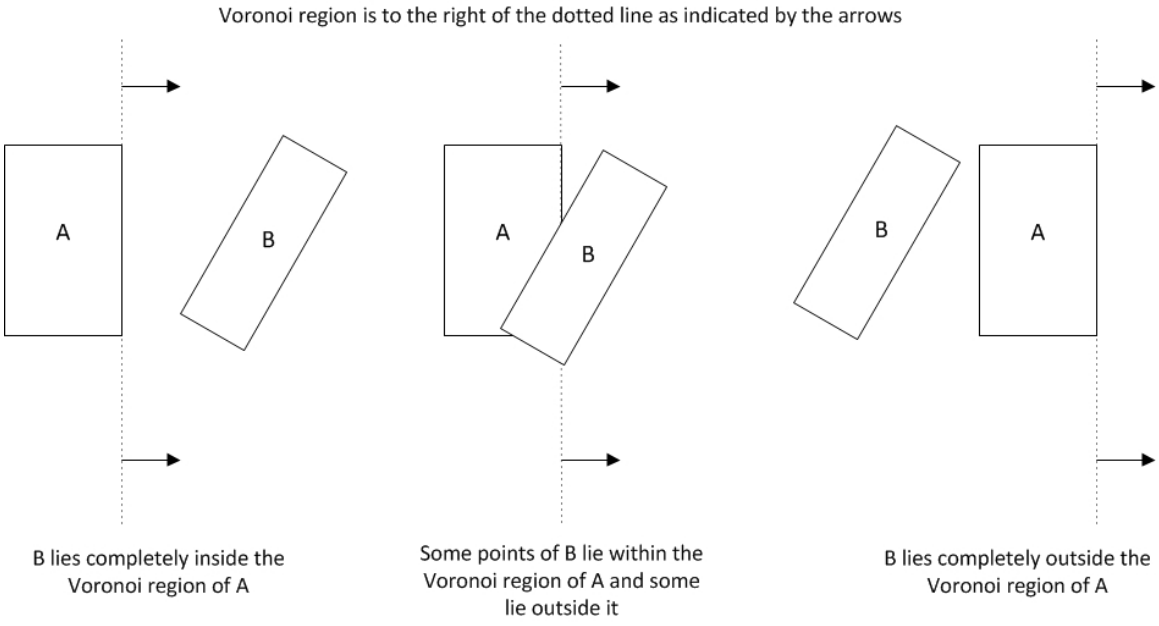

Fig. 4.9: Use of external Voronoi regions to determine configuration of pair of edges

To determine if primitives are overlapping or separate, a modification of the separating axis theorem used in stage one for OBB comparison (Gottschalk et al. 1996) is used. Primitives are projected onto a unit direction and the distance between intervals is computed. This value represents the lower bound for the actual separation distance. If one

of the closest points is in the interior of a primitive, the maximum magnitude of the normal vector to each primitive gives the upper bound of the separation distance. If both the lower and upper bound values are zero, then the primitives are overlapping.

The algorithm described in Larsen et al. (1999) uses optimization techniques to speed up computations and by pruning nodes sooner in the tree traversal. It optimizes proximity queries through a technique called priority directed search. Priority queue is a variation of the conventional queue data structure in which elements have a user-determined priority associated with them. Elements with higher priority are given preference over elements with lower priority. The priority in this case decides the order in which proximity tests are to be performed. The algorithm for proximity queries assigns priority based on the distance from the current BV pair. Closer BV pairs are given higher priority and checked prior to BVs lying farther away.

Another technique used to optimize proximity queries takes advantage of coherence between successive frames of motion. The distance between a pair of objects changes by relatively small amounts between successive frames due to the high frequency of performing proximity queries. The closest pair of triangles from a given query is recorded or cached, and their corresponding separating distance is used to initialize ‘D’ rather than using a very large value as shown in Figure 4.7. This technique is known as triangle caching. As the distance moved is very small, the value of ‘D’ from a prior frame is very close to the current ‘D’ between a pair of objects. Details of the priority directed search and triangle caching optimizations can be found in Larsen et al. (1999).

4.6. PROTOCOL

Figure 4.10 shows a real-time simulation framework to assist excavation operators in carrying out operations in the presence of buried utilities, and thus in avoiding unintended strikes. The framework is generic and can be used to simulate any construction operation where operators experience reduced visual guidance and spatial awareness. SeePlusPlus is the 3D visualization component of the framework that offers users a virtual, interactive view of the ongoing operation. The S2A2 (Sensor Stream Acquisition Allocation) Framework is responsible for transmitting position and orientation sensor data from the real-world jobsite into the virtual world. Thus input from S2A2 is used to update 3D models in the visualization and real-time spatial queries.

There are dedicated modules for the creation of static and dynamic 3D entities. The B3M Toolkit is used to create georeferenced 3D models of buried utilities. These models can be directly used in the simulation due to being location-aware. VirtualWorld Creator provides 3D models representing articulated construction equipment used in the operation. While Figure 4.10 shows an excavator being introduced, it can provide the appropriate 3D equipment model based on the type of operation being simulated. PROTOCOL is the module responsible for providing spatial queries by using 3D models representing static and dynamic entities, using sensor input from the real world, and providing audio-visual warning feedback to the user for accident avoidance.

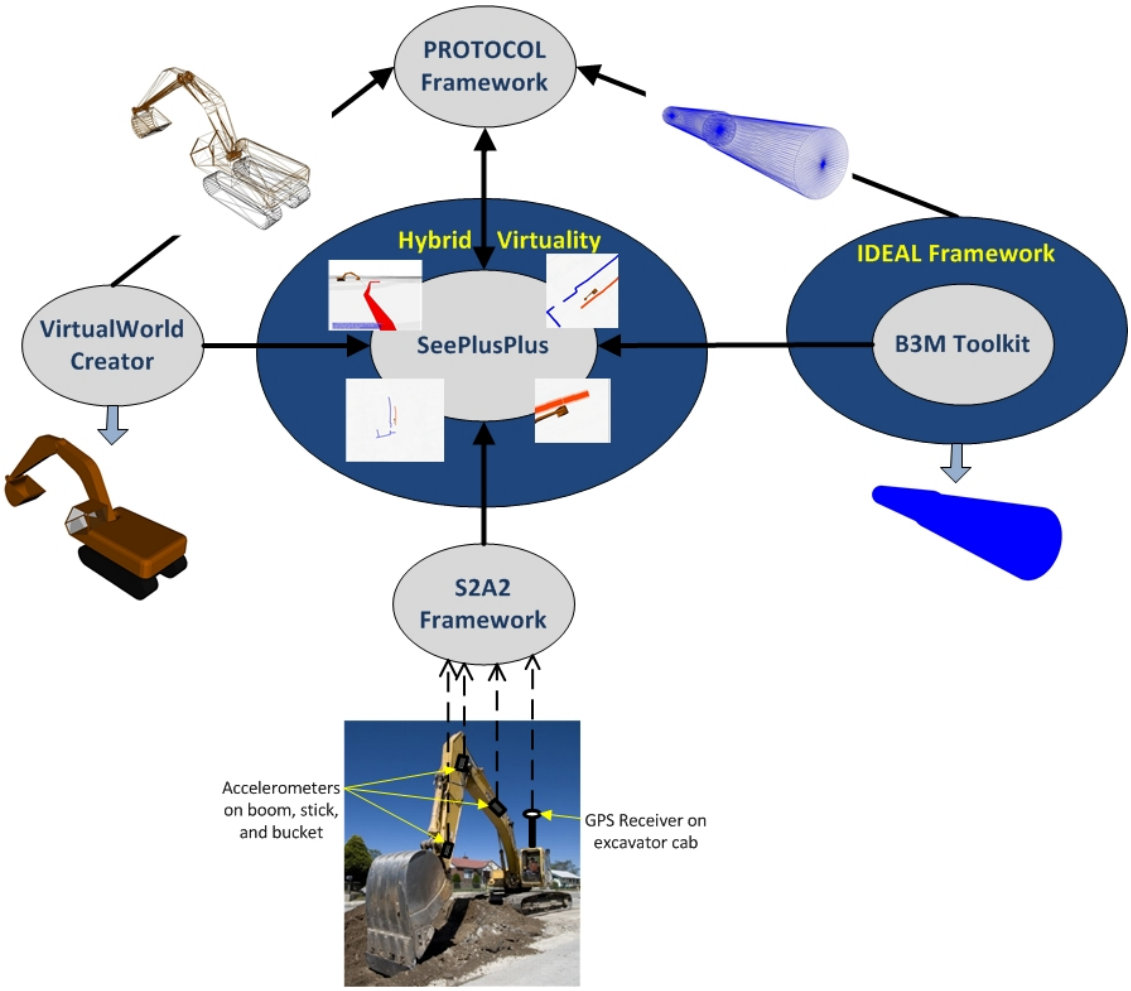

Fig. 4.10: Constituent modules making up the overall framework to assist excavation operators in avoiding buried utility strikes

Thus PROTOCOL is a software module that is designed to integrate with a 3D visualization system and allow users to make spatial queries. Users can select between Proximity Monitoring, Collision Detection, and Tolerance Checking queries. The PROTOCOL name is chosen to reflect the three categories of queries available. PROTOCOL's geometric computational functionality is designed using the Proximity Query Package (PQP) library (PQP Gamma 1999). The PQP library uses the geometric

primitives that make up objects, as well as objects' global poses, to compute the proximity between two or more objects.

As described in Section 4.5, the use of low-level geometric primitives for distance computation and collision detection ensures that these computations can be made to a much higher resolution than what would be possible through the use of bounding volumes alone. However, the use of polygon surface models to represent static and dynamic entities means that 3D objects need to be decomposed into their primitives in order to be used by the algorithms. The decompositions represent a polygon surface model as a list of its constituent triangle primitives; the decompositions are stored in memory as 'tris' files. A 3D model composed of quads or other non-triangular primitives is also automatically decomposed into its equivalent list of triangles. The geometric primitives representing a rigid body remain consistent throughout its operation. Thus pose updates from real-world sensors, when combined with tris files, can represent a real-world entity's pose in a virtual world. Every tris file can be used for multiple instances of the same 3D model by simply substituting each instance's position-orientation.

PROTOCOL is implemented as a plugin to SeePlusPlus and is presented to the user as a Graphical User Interface (GUI) where queries can be created between entities. Figures 4.11 and 4.12 show the PROTOCOL GUI for static query creation, and dynamic query creation, respectively. One-one and one-many relationship queries can be created. Any combination of distance, collision, and/or tolerance queries can be instantiated between

two entities. Tolerance queries can be used to check if a predetermined safety threshold distance has been breached during operation.

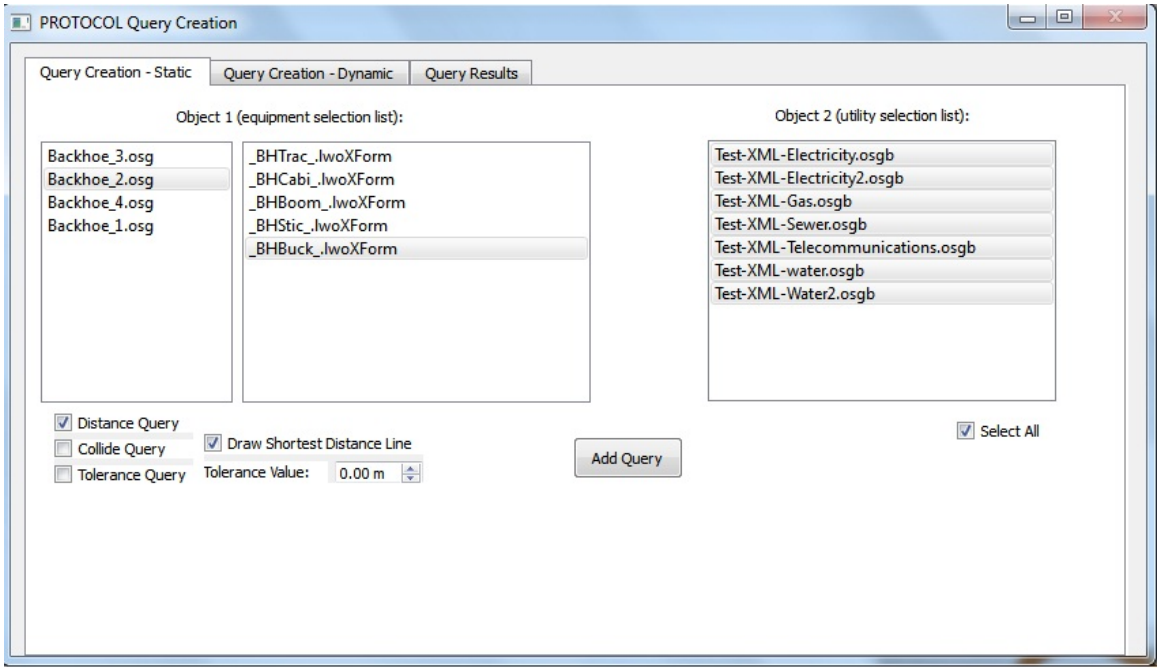

Fig. 4.11: PROTOCOL Graphical User Interface for spatial query instantiation between dynamic and static entities

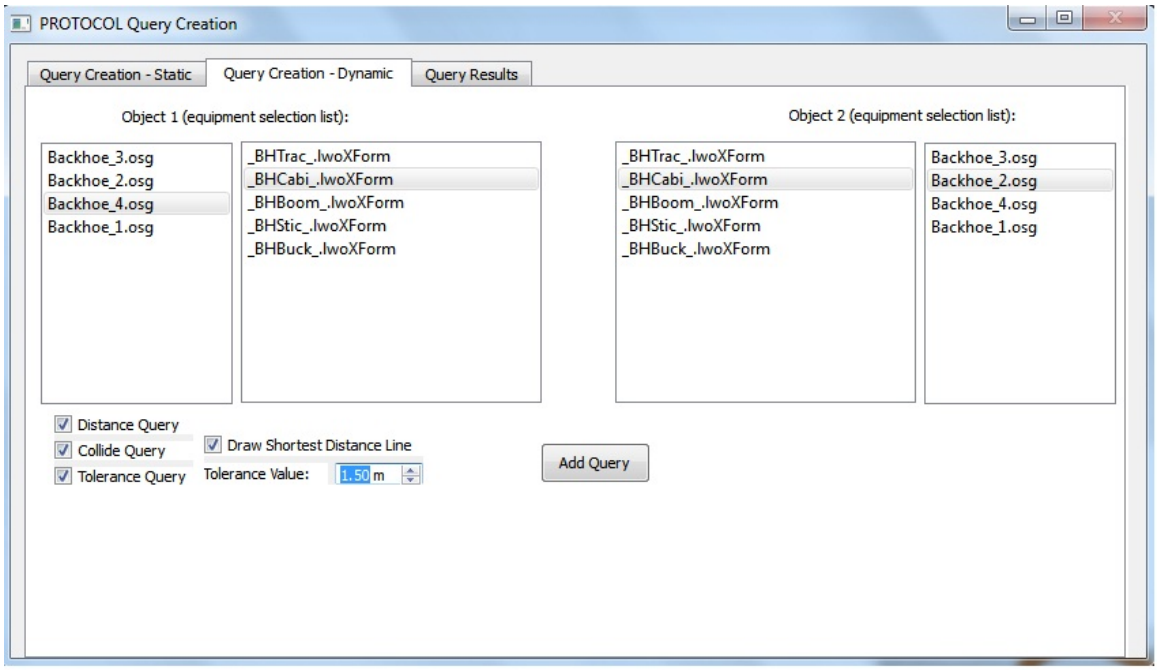

Fig. 4.12: PROTOCOL Graphical User Interface for spatial query instantiation between dynamic entity pairs

Figure 4.13 shows the list of queries created and entities participating in a given query. The GUI is designed as a tabbed widget, thus enabling users to switch easily from one view to the next. The GUI also provides an option to render a line joining the closest points between a pair of objects; this line is meant to assist operators in identifying the entities that are participating in the query when the number of entities is high. In addition, queries created in PROTOCOL can be saved and re-used for future use. This feature is conceived as being useful in scenarios where the end-user, such as an equipment operator, can load a pre-created virtual scene and proximity queries without having related expertise in those areas.

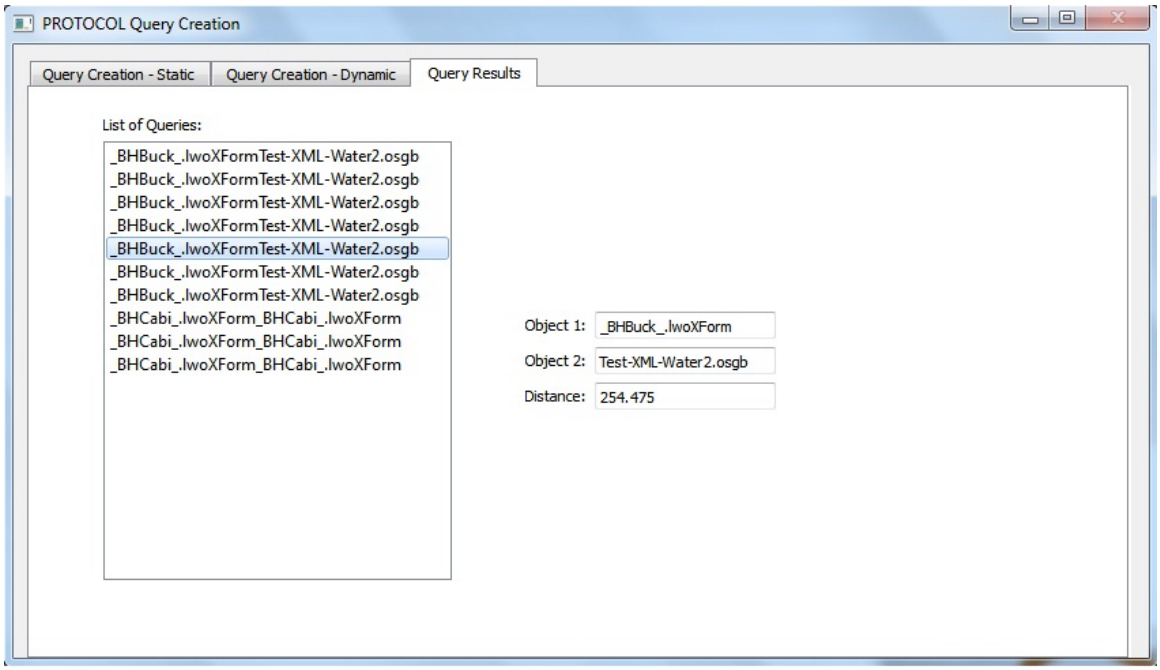

Fig. 4.13: PROTOCOL Graphical User Interface for displaying list of queries created and entities involved in the query

4.7 Validation Experiments

The two key parameters to measure a proximity monitoring framework's effectiveness are measurement error and latency. Measurement error in this case refers to the error of the computed distance with regard to the ground truth or theoretically expected values. Latency in this case is a measure of time lag between an event occurring in the real world and a proximity monitoring framework providing output to warning systems that end-users depend upon. Testing details of these facets requires the design of appropriate experiments. This section describes experiments carried out to test the measurement error and latency of the PROTOCOL module. The experiments were carried out at the Intelligent Sensing and Automation Testbed (ISAT) at the National Institute of Standards

and Technology (NIST) in Gaithersburg, Maryland. The following sections describe the details of the experimental setup, methodology, and results.

4.7.1 Proximity Test

PROTOCOL's proximity measurement performance is evaluated in this experiment. The metric for PROTOCOL's proximity measurement performance is the difference between PROTOCOL's reported proximity and the ground truth or theoretically expected values. Any variations between computed and expected values will demonstrate PROTOCOL's contribution to computational errors. However, in order to ensure that any errors are purely computational and do not originate from position and orientation tracking, a tracking system with sufficient accuracy must be used. Although modern, commercial GPS units are capable of sub-centimeter position accuracies using certain technologies—such as differential-GPS (DGPS) and real-time kinematic GPS (RTK-GPS)—their accuracies can be reduced due to environmental factors, surroundings, and line-of-sight to the sky (Lodha et al. 2002). Thus the use of outdoor GPS or technologies that cannot guarantee a consistently high level of positional accuracy can adversely affect the computations performed by PROTOCOL. If position tracking is inaccurate and the magnitude of uncertainty cannot be measured, the difference between PROTOCOL-computed distances and theoretically expected values cannot be attributed to either positional input or PROTOCOL computations.

In order to ascertain the cause of uncertainty, a tracking system capable of consistent readings was required for this experiment. Indoor GPS (iGPS) is one such tracking

technology that is capable of sub-millimeter (± 0.250 mm) position tracking accuracy (Nikon Metrology 2012). iGPS can be compared to the Global Positioning System's constellation of satellites in which each indoor transmitter plays a similar role to a GPS satellite. Instead of satellites, iGPS uses infrared laser transmitters that emit laser pulses. When a receiver obtains pulses from more than a single transmitter, that receiver can compute its position and orientation within the coordinate system defined by the transmitters. Photo detectors pick up the signals and compute angles and positions based on the timing of the arriving light pulses. Figure 4.14 shows the instruments and sensors used to conduct the experiment. Figure 4.15 is a graphical, overhead view of the ISAT test-bed.

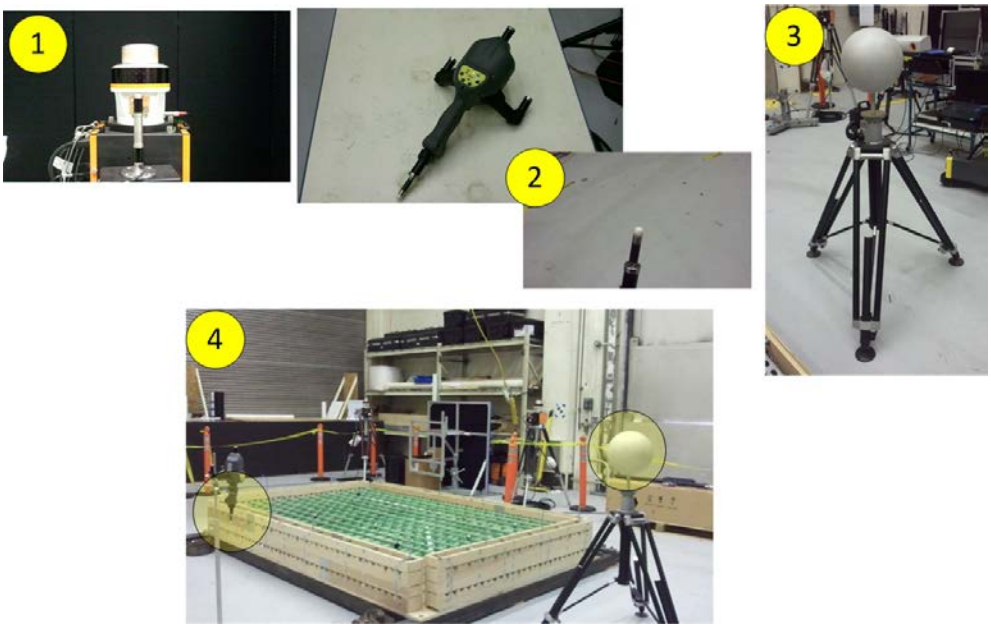

Fig. 4.14: 1- Indoor GPS Transmitter; 2- Indoor GPS tracking probe (i6 Probe) and its tracking tip; 3-Test Sphere mounted on tripod; 4-Experiment setup with i6 Probe placed adjacent to Test Sphere

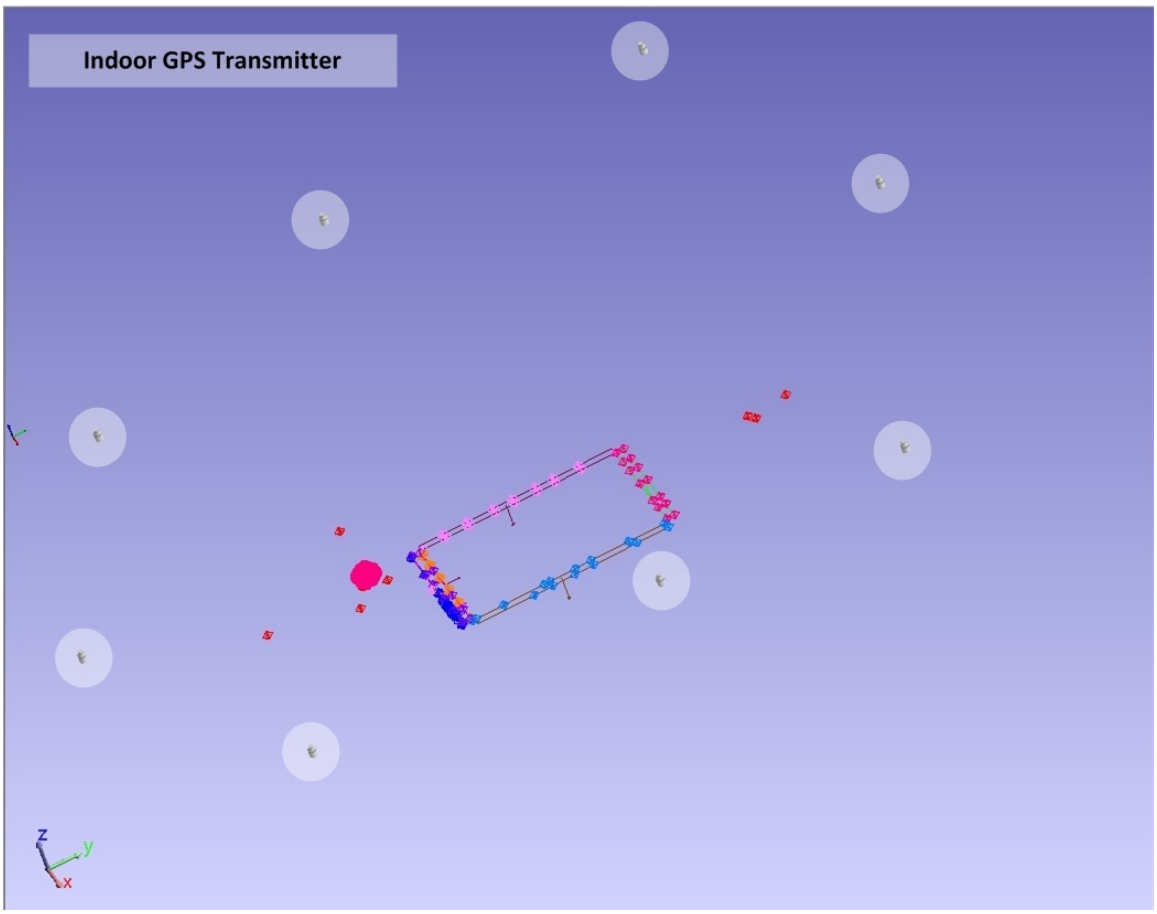

Fig. 4.15: Graphical representation of the layout of the eight iGPS transmitters (large circles) in the ISAT with some of the measured data shown as dots in the figure

The radius and center of the Test Sphere in Figure 4.14(3) was measured using the iGPS. Distances measured with the iGPS were treated as the ground truth. The Test Sphere and probe tip were modeled as polygon surface spheres to be used in a geometric proximity analysis and real-time 3D visualization. The proximity test consisted of a series of 7 iterations starting at a distance of less than 0.3048 meters (1 foot) surface-to-surface separation between the Test Sphere and the probe tip. Successive iterations were made at 0.6096 meters (2 feet) increments. For each iteration, 10 readings were made to account

for variations in the iGPS tracking. For every measurement, the error (i.e., the difference between the value computed by PROTOCOL and the ground truth) was calculated. Table 4.4 shows the separation distance error for the complete range of separation distances. Since the iGPS positional uncertainty for a given point is ±0.250 mm, the difference between two iGPS points can vary by a combined ±0.354 mm, where $0.354 = \sqrt{(2 * (\pm 0.25^2))}$.

Iteration #	Nominal Separation Distance meters (feet)	Ground Truth (Average for 10 readings) (mm) ± 0.354 mm	PROTOCOL Computed (Average for 10 readings) Distance (mm)	Δ (mm) [Computed – Ground Truth]
1	< 0.305 (1)	290.360	292.420	1.560
2	0.305 – 0.914 (1 – 3)	724.730	725.878	1.148
3	0.914 – 1.524 (3 – 5)	1188.768	1189.552	0.784
4	1.524 – 2.134 (5 – 7)	1856.836	1857.434	0.598
5	2.134 – 2.743 (7 – 9)	2475.590	2476.090	0.500
6	2.743 – 3.353 (9 – 11)	3032.077	3032.490	0.413
7	> 6.096 (20)	7073.196	7073.520	0.324

Table 4.4: Comparison of PROTOCOL-computed values and ground truth values for various separation distances

4.7.2 Latency Test

A good proximity monitoring system must be capable of analyzing input and providing real-time or near real-time output in order for it to be useful. Bryson (1993) identified the two main causes of dynamic distortion in real-time computer graphics systems as lag and frame rate. In these experiments, lag was defined as the time offset in the data stream from a tracker to the graphical image. It was also observed that the frame rate depended upon the scene complexity, varying from very simple renderings to those requiring considerable computation, and renderings in response to user movement (Bryson 1993).

In its basic form, latency is a measure of the overall delay in a system. However its constituent time delays and their measurements are system- and implementation-specific. In the case of PROTOCOL, the delay is measured between an event occurring in the real world and a warning being given to the user by the system. For example, if PROTOCOL is set to a tolerance mode of one meter, the system must provide output for audio/visual warnings when two entities are within one or fewer meters of each other. The time delay between entities coming within the tolerance distances in the real world and audio/visual warnings being provided to the user is an example of overall system latency, as represented in Figure 4.16.

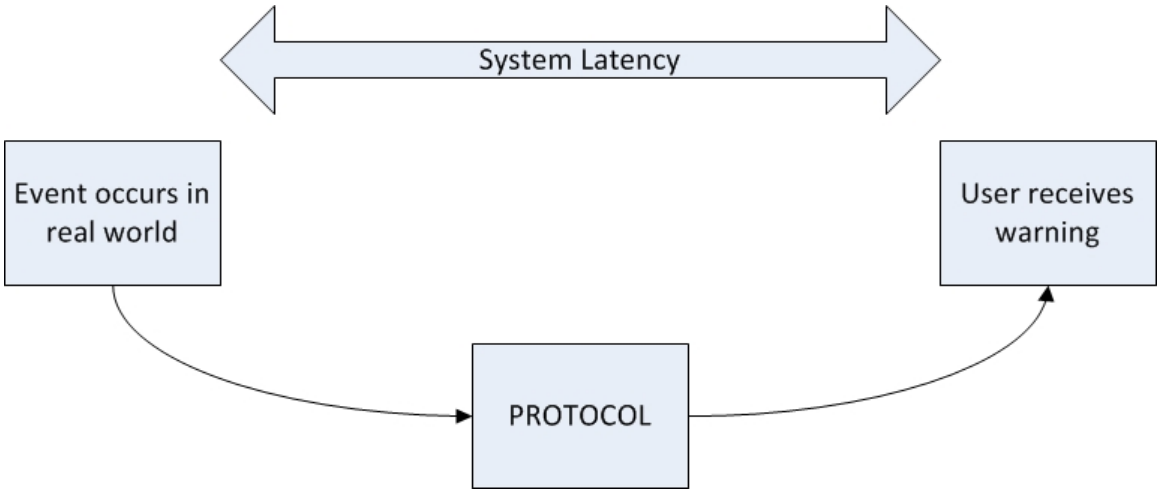

Fig. 4.16: Time lag between an event occurring in the real world (e.g., jobsite) and the user receiving a warning is defined as system latency

System latency in relation to PROTOCOL is further sub-divided into Incoming External Latency, Internal Latency, and Outgoing External Latency based upon the time taken to complete specific tasks, as shown in Figure 4.17. Incoming External Latency refers to the time taken for sensors placed on or around equipment to register a change in position-orientation and transmit it to PROTOCOL. Incoming External Latency is a function of the data transmission medium chosen and the refresh rates of the sensors. Internal Latency is the time taken by PROTOCOL to update the existing graphical database and perform a geometric analysis. Hence internal latency is a function of scene complexity and the number of entities being modeled, and the number and types of queries being performed. Finally, Outgoing External Latency is the time taken to transmit the analysis output to the audio/visual systems that process and supply appropriate warnings to the user. It follows that Outgoing External Latency is a function of the data transmission medium and the types of warning devices being implemented. A visual warning

mechanism that is part of the visualization framework can reduce the time to negligible values, while disparate warning devices (e.g., audible alarms) can add additional time due to additional transmission being involved.

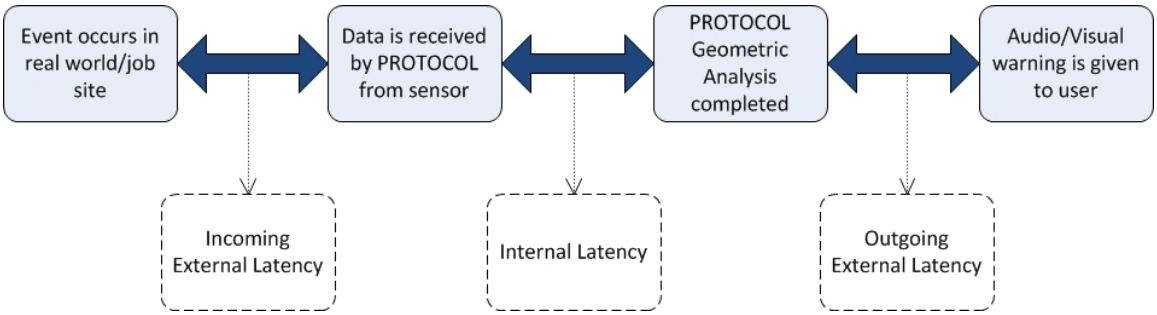

Fig. 4.17: Breakdown of system latency into constituent components

In this test, a tolerance trigger was used to measure the latency of the system. The experiment setup was identical to that of the distance test, including the i6 Probe, Test Sphere, and iGPS transmitter layout. The test was designed to trigger a visual warning whenever the probe tip was within 30.48 cm (1 foot) of the Test Sphere (see Figure 4.18), as measured by PROTOCOL.

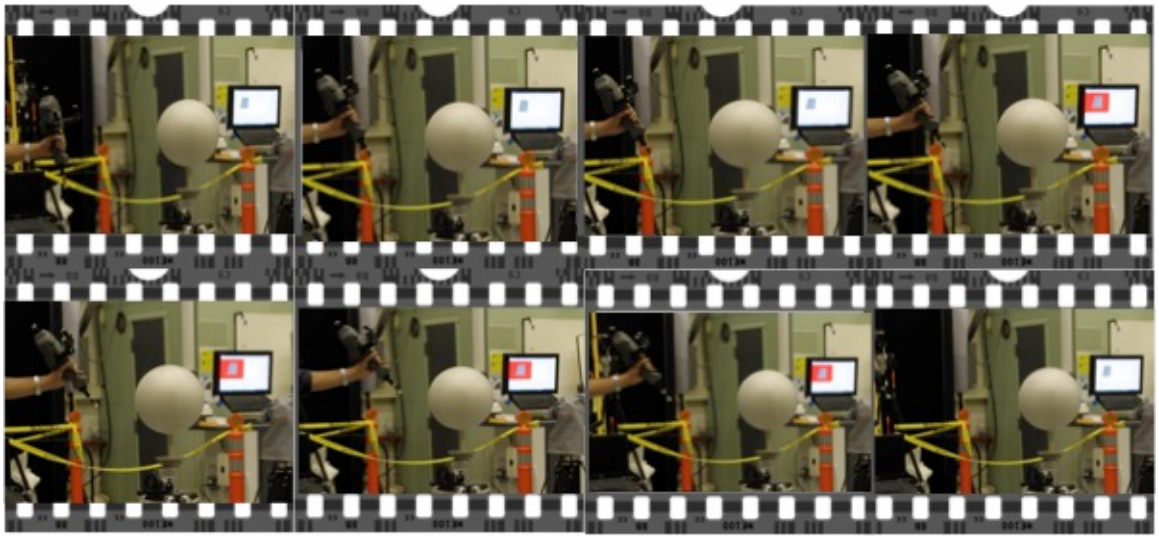

Fig. 4.18: Image captured from a video recording of the latency test showing change in background screen color from white to red, signifying breach of preset safety threshold

Four iterations of the latency test were performed, and the three constituent latencies making up overall latency were recorded, as shown in Table 4.5. TCP/IP socket connections were used to transmit pose data from the iGPS server to PROTOCOL's input stage. Due to the high frequency of streaming data it was critical to ensure that time durations of incoming, internal, and outgoing segments were computed accurately by recording time stamps along with the pose coordinate for every packet of data. Thus an individual pose coordinate's progress from the iGPS server through the entire visualization system could be tracked without error. Incoming external latency was measured as the time offset between a pose data point being streamed by the iGPS server and being received by the visualization system. Internal latency was measured as the time between a data point being received and its proximity analysis being completed. Outgoing external latency was measured as the difference in time stamps between

analysis being completed (and warning being detected) and the visualization system acknowledging the warning condition by altering the screen color from white to red.

Iteration #	Incoming External Latency (milliseconds)	Internal Latency (milliseconds)
1	15.4	13
2	8.0	20
3	3.3	18
4	8.1	10

Table 4.5: Latency test results for PROTOCOL. The Outgoing External Latencies were negligible (0 milliseconds from recorded time stamps) and are not shown

In addition to the above, the iGPS system has an associated latency due to the frequency at which it updates readings. The iGPS system has an update rate of 40 Hz, which implies that a data point transmitted by it has an associated latency of 25 ms. The incoming external latency values presented in Table 5 do not include the 25 ms latency. During the experiments, the probe was being moved at an average speed of 0.54m/s. As the iGPS system updates the location at 40 Hz, the probe tip's reported position can be up to 13.5 mm behind its actual position in the real world. For a given sensor and its corresponding dynamically tracked entity, the offset would increase with greater speeds of motion and/or reductions in the position update frequency. In order to account for this offset between reported and actual positions, the future work of the authors includes a stochastic

approach to predict warnings in advance, as introduced in the Conclusions and Future Work section.

The Nagle algorithm that coalesces smaller data packets into fewer larger packets is understood to be one of the causes for transmission delay in the case of incoming external latency (Nagle 1984). Investigation of the data stream from the server application showed that a similar coalescing of data points was occurring with multiple data points having the exact timestamp. As a result of multiple data points being sent in bursts, the time offset for some of the incoming external latencies was observed to be very high, and were ignored for the purposes of mean and standard deviation calculations. In the current implementation of the visualization system, the incoming sensor stream updates are associated with the update traversals of the overall visualization system.

As a result, the sensor stream data has some additional latency due to it residing in the client socket's buffer prior to being processed. A modification of the data transmission design to reduce latency for server and client ends is part of the authors' ongoing research. Internal Latency times showed a relatively consistent value for analyzing the graphical database consisting of the Test Sphere and the probe tip, as seen in Figure 4.19(1, 2). The Outgoing External Latency was negligible since the 3D visualization itself was used to provide visual warnings, and is thus not represented in the results displayed in Table 5.

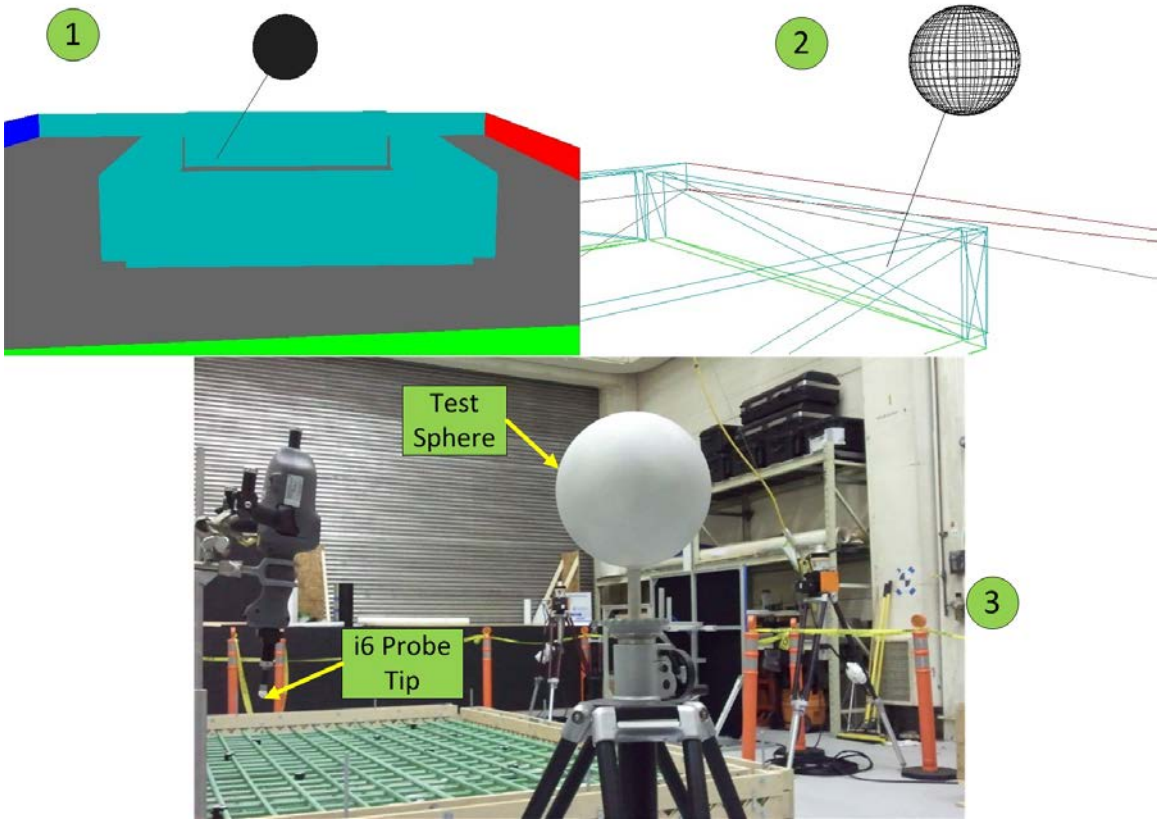

Fig. 4.19: (1) Real-time 3D visualization of experiment; (2) Polygon surface models used for graphical analysis; (3) Experiment setup with Test Sphere and i6 Probe tip

4.7.3 Discussion of the Experimental Results

The distance test results show that the mean separation distance error with the standard deviation is 0.761 ± 0.446 mm. Figure 4.20 shows a plot of the separation distance error as a function of the ground truth separation distance. The trend of increasing error as the separation distance decreases is potentially due to the position of the iProbe during the tests. In the case where the Test Sphere and iProbe were in very close proximity, the receivers on the iProbe were partially blocked by the Test Sphere, resulting in small variances of the reported position. As the reported position of the iProbe tip's center is

used to place its corresponding 3D model, variances of the reported positions led to unintended translations in the 3D model.

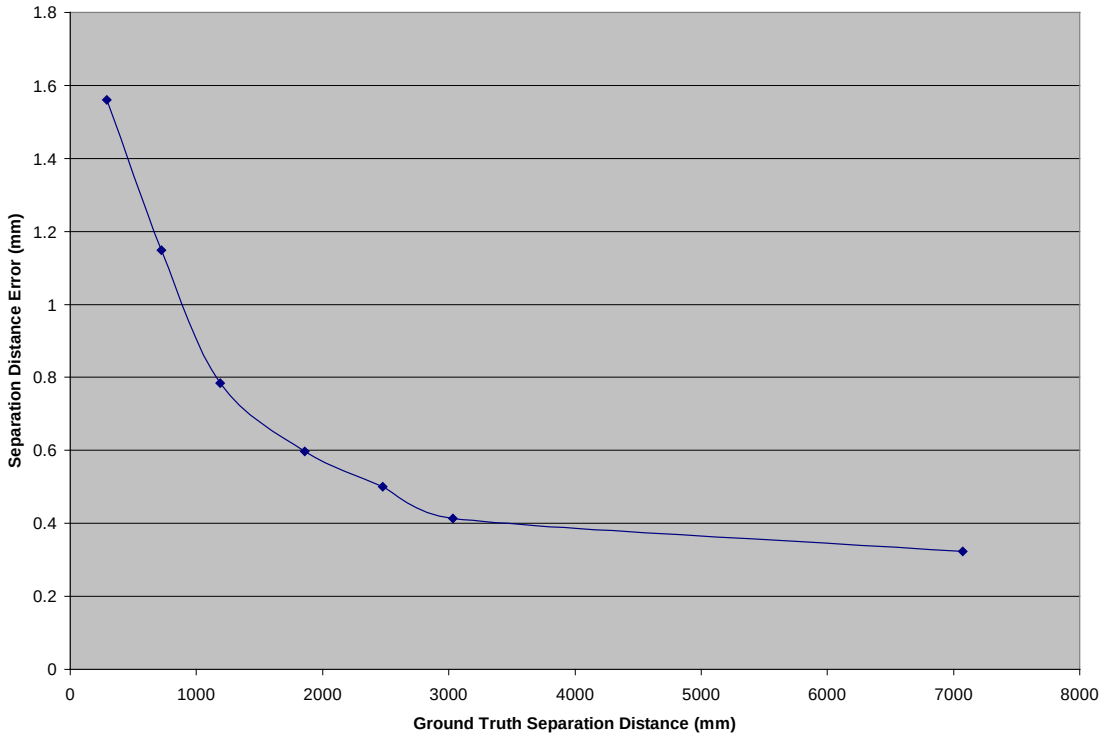

Fig. 4.20: Separation Distance Error as a function of Separation Distance

The latency test showed that the mean Incoming External Latency with the standard deviation was 8.7 ± 4.99 ms. Mean Internal Latency with the standard deviation was 15.25 ± 4.57 ms. The observed values of Incoming External and Internal Latencies yielded frame rates in the range of 15 to 24 frames per second (fps) or Hz, as observed in the OpenSceneGraph-based visualization. The frame rate did not yield any noticeable lag in operation and also provided fps values that lie in the minimum acceptable range of 15–30 Hz for real-time rendering, giving smooth continuous motion (Bowman 2008, Malhotra 2002, Wloka, M. and Zeleznik 1996).

4.8 Conclusions and Future Work

In this chapter the authors investigated the types of spatial conflicts occurring between entities on construction jobsites, and identified limitations in current methodologies in dealing with scenarios that involve concealed or buried infrastructure. We presented a computing framework—based on position-orientation sensors' input and 3D virtual models—that provides distance, collision, and tolerance queries. The computation framework is designed to be able to work with a real-time 3D visualization scheme that provides equipment operators with job-site views that are not possible through conventional on-site cameras. Results of proximity measurement performance and latency tests carried out at the National Institute of Standards and Technology's Intelligent Sensing and Automation Testbed were presented.

The future goals are to introduce a stochastic model in the warning mechanism. The current implementation is a reactive approach that warns users of impending collisions when entities cross safety thresholds in the real world. However, the impact of data transmission times from sensors to the on-board computation center can result in unfavorable latency. This can be averted through a predictive approach in which an object's current motion trajectory and speed is used to project when a safety threshold will be breached, and warnings can be given further in advance without being affected by high transmission times to the same extent.

4.9 **Acknowledgements**

The presented research was partially funded by the US National Science Foundation (NSF) via Grants CMMI- 825818 and CMMI-927475. The writers gratefully acknowledge NSF's support. Any opinions, findings, conclusions, and recommendations expressed in this chapter are those of the authors and do not necessarily reflect the views of the NSF or the University of Michigan.

4.10 References

Arditi, D., Ayrancioglu, M. and Shu, J. (2005). “Worker safety issues in nighttime highway construction”, Engineering Construction and Architectural Management, 12(5), pp. 487–501.

Bajaj, C., Wyvill, B., Bloomenthal, J., Guo, B., Hart, J., Wyvill, G. (1996). “Implicit Surfaces for Geometric Modeling and Computer Graphics”, SIGGRAPH Lecture 1996, <http://www.cs.princeton.edu/courses/archive/fall02/cs526/papers/course11sig96.pdf> (06/17/2012).

Bowman, D. (2008). “Real-time 3D graphics for VEs”, <http://courses.cs.vt.edu/~cs5754/lectures/rtgraphics.pdf> (07/26/2012).

Bryson, S. (1993). “Effects of Lag and Frame Rate on Various Tracking Tasks”, In Proceedings of SPIE Stereoscopic Displays and Applications, volume 1915, Sept. 1993.

Bureau of Labor Statistics (2012). “Census of Fatal Occupational Injuries” <http://www.bls.gov/iif/oshcfoi1.htm#2009> (06/17/2012).

Chen, J. Y.C., Haas, E. C., and Barnes, M. J. (2007). “Human performance issues and user interface design for teleoperated robots”, IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, 37: pp. 1231–1245.

Cheng, T., and Teizer, J. (2011). “Crane Operator Visibility of Ground Operations”, 2011 Proceedings of the 28th ISARC, Seoul, Korea, pp. 699–705.

Chi, S., Caldas, C., and Gong, J. (2008). “A crash avoidance framework for heavy equipment control systems using 3D imaging sensors”, ITCON, 13, pp. 118–133.

Cho, Y., Haas, C., Liapi, K., Sreenivasan, S.V. (2002). “A framework for rapid local area modeling for construction automation”, Automation in Construction, 11 (6) (2002), pp. 629–641.

Dana, P. H. (2000). “Unit 017 – Global Positioning System Overview”, <http://www.ncgia.ucsb.edu/giscc/units/u017/u017.html> (06/17/2012).

Dasgupta, S., and Banerjee, A. (2006). “An augmented-reality-based real-time panoramic vision system for autonomous navigation”. IEEE Trans. Systems, Man and Cybernetics, Part A, vol. 36, no 1, pp. 154–161, 2006.

Durikovic, R. (2012) "Lecture 13: Representation of Solids" <http://www.sccg.sk/~durikovic/classes/MRT/3DModeling3%20Representation.pdf> (06/17/2012).

Federal Highway Administration (FHWA) (2012). “Worker Safety”, <http://www.ops.fhwa.dot.gov/wz/workersafety/index.htm> (06/17/2012).

Foley, J. D., van Dam, A., Feiner, S.K., and Hughes, J. (1992). "Computer Graphics: Principles and Practice", Addison Wesley, Second Edition, 1992, pp. 533–562

Geokov (2012). “UTM – Universal Transverse Mercator Projection”, <http://geokov.com/Education/utm.aspx> (06/17/2012).

Gottschalk, S., Lin, M. C., Manocha, D. (1996) “Obbtree: A hierarchical structure for rapid interference detection”, In SIGGRAPH ’96 Proc., 1996.

Hearn, D., and Baker, M.P. (1994). "Computer Graphics", Prentice Hall Second Edition, 1994, pp. 355–362

Hine III, B. P., Stoker, C., Sims, M., Rasmussen, D., Hontalas, P., Fong, T. W., Steele, J., Barch, D., Andersen, D., Miles, E., and Nygren, E. (1994). “The Application of

Telepresence and Virtual Reality to Subsea Exploration", *The 2nd Workshop on Mobile Robots for Subsea Environments, Proc. ROV'94*, May, 1994.

Hinze, J. W., and Teizer, J. (2011). "Visibility-related fatalities related to construction Equipment", Safety Science 49 (2011), pp. 709–718.

Huang, X., Bernold, L. (1997). "CAD integrated excavation and pipe laying", Journal of Construction Engineering and Management, 123 (3) (1997), pp. 318–323.

Huber, D., Herman, H., Kelly, A., Rander, P., and Warner, R. (2009). "Real-time Photorealistic Visualization of 3D Environments for Enhanced Teleoperation of Vehicles", Proceedings of the 2nd International Conference on 3D Digital Imaging and Modeling, Kyoto, Japan.

Hwang, S. (2011). "Ultra-wideband technology experiments for real-time prevention of tower crane collisions", Automation in Construction, Volume 22, March 2012, pp. 545–553.

Kamat, V.R., and Martinez, J.C. (2007). "Interactive collision detection in three-dimensional visualizations of simulated construction operations," Engineering with Computers.Vol.2,2007.

Kamat, V. R., and Martinez, J. C. (2005). "Dynamic 3D Visualization of Articulated Construction Equipment", Journal of Computing in Civil Engineering, Vol. 19, No. 4, American Society of Civil Engineers, Reston, VA, 356–368.

Kim, C.W., Haas, C.T., Liapi, K.A., Caldas C.H. (2006). "Human-assisted obstacle avoidance system using 3D workspace modeling for construction equipment operation", Journal of Construction Engineering and Management, 20 (3) (2006), pp. 177–186.

Kini, A.P., King, K., and Kang S., (2009). “Sensor Calibration and Real-time Tracking of a Backhoe Loader Using the iGPS System”, Quality Digest Magazine, <http://www.qualitydigest.com/inside/cmsc-article/sensor-calibration-and-real-time-tracking-backhoe-loader-using-igps-system.html> (06/17/2012).

Kwon, S., Bosche, F., Kim, C., Haas, C. T., and Liapi, K. (2004). “Fitting range data to primitives for rapid local 3D modeling using sparse range point clouds.” *J. Autom. Constr.*, 13_1, pp. 67–81.

Larsen, E., Gottschalk, S., Lin, M. C., Manocha, D. (1999) “Fast Proximity Queries with Sphere Swept Volumes”, Available online as technical report TR99-018, Department of Computer Science, UNC Chapel Hill <http://gamma.cs.unc.edu/SSV/ssv.pdf> (06-22-2011).

Lin, M.C., Gottschalk, S. (1998), “Collision detection between geometric models: a survey”, <ftp://ftp.cs.unc.edu/pub/users/lin/cms98.pdf> (11/19/2011).

Lodha, S., Charaniya, A. P., Faaland, N. M., and Ramalingam, S. (2002). “Visualization of spatiotemporal GPS uncertainty within a GIS environment”, In Proceedings of SPIE Conference on Radar Sense Technology and Data Visualization, April 2002, pp. 216–227.

MacCollum, D.V. (1995). “Construction Safety Planning”, John Wiley & Sons, New York, pp. 53–54

Malhotra, P. (2002). “Issues involved in real-time rendering of virtual environments”, Master’s thesis, Faculty of Virginia Polytechnic, 2002.

McHenry, K., Bajcsy, P. (2008). “An Overview of 3D Data Content, File Formats and Viewers”, Technical Report ISDA08-02, 2008.

Nagle, J. (1984). "Congestion Control in IP/TCP Internetworks", RFC-896, Internet Engineering Task Force, January 1984. <http://tools.ietf.org/pdf/rfc896.pdf> (07/26/2012).

Nikon Metrology Inc. (2012) "iGPS" <http://www.nikonmetrology.com/en_US/Products/Large-Volume-Applications/iGPS/iGPS/(brochure)> (06/17/2012).

Occupational Safety and Health Administration (2010) <http://www.osha.gov/pls/oshaweb/owadisp.show_document?p_table=INTERPRETATIONS&p_id=27308 > (06/17/2012).

Oloufa A., Ikeda, M., and Hiroshi O. (2002). "GPS-Based wireless collision detection of construction equipment", International Symposium on Automation and Robotics in Construction, 19th (ISARC). Proceedings. National Institute of Standards and Technology, Gaithersburg, Maryland. September 23-25, 2002, pp.461–466.

PHMSA (2012), "Significant Pipeline Incidents through 2010 By Cause" <http://primis.phmsa.dot.gov/comm/reports/safety/SigPSIDet_2001_2010_US.html?nocache=9140#all> (01/11/2012).

PQP Gamma (1999) "PQP – A Proximity Query Package" <http://gamma.cs.unc.edu/SSV/> (6/22/2011).

Preston, R., Doane, J., and Kiwio, D. (2005). "Using real world data to create OPNET Models DRAFT", The MITRE Corporation, <http://www.mitre.org/work/tech_papers/tech_papers_05/05_0850/05_0850.pdf> (06/17/2012).

Ruff, T.M. and Hession-Kunz, D. (2001). “Application of Radio-Frequency Identification Systems to Collision Avoidance in Metal/Nonmetal Mines”, IEEE Transactions on Industry Applications, 2001. 37(1).

Schiffbauer W.H. and Mowrey G.L. (2001). “An environmentally robust proximity warning system for hazardous areas”, Proceedings of the ISA Emerging Technologies Conference, Triangle Park, NC. Vol. 2091, p10.

Shen C., O'Brien J.F., Shewchuck, J.R. (2004). “Interpolating and approximating implicit surfaces from polygon soup”, *ACM Trans. Graph.*, 23, 3 (2004), pp. 896–904.

Son, H., Kim, C., Choi, K. (2010). “Rapid 3D object detection and modeling using range data from 3D range imaging camera for heavy equipment operation”, Automation in Construction, vol. 19, no. 7, November 2010, pp. 898–906.

Son, H., Kim, Ch., Kim, H., Choi, K.-N., 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 (2008), pp. 232–237.

Talmaki, S. A., and Kamat, V. R. (2012), “Real-Time Hybrid Virtuality for Prevention of Excavation Related Utility Strikes”, Journal of Computing in Civil Engineering, American Society of Civil Engineers, Reston, VA. (In Review)

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

Teizer, J., Allread, B.S., Fullerton, C.E., Hinze, J., (2010b). Autonomous pro-active realtime construction worker and equipment operator proximity safety alert system. Automation in Construction 19 (5), pp. 630–640.

Teschner, M., Heidelberger, B., Manocha, D., Govindaraju, N., Zachmann G., Kimmerle S., Mezger J., Fuhrmann A. (2005). “Collision Handling in Dynamic Simulation Environments”, In Eurographics Tutorials 2005, pp. 79–185.

Wloka, M. and Zeleznik, R.C. (1996). “Interactive real-time motion blur”, Visual Computer, 1996, pp. 273- 295.

Zhang C., Hammad, A., 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, 14, pp. 204–228.

Chapter 5

Evaluation of Sensor Retrofits for Real-time Graphical Emulation of Articulated Construction Equipment

5.1 Introduction

In fields such as mining, quarrying, and construction, equipment monitoring plays a crucial role in accident prevention (MSHA 2012). Construction jobsites, in particular, are occupied by workers and equipment, often belonging to different sub-contractors (Castro-Lacoutere et al. 2007). The visibility available to an operator on a dynamic construction site can often be blocked by various obstacles such as materials, temporary or permanent facilities, other equipment, and even workers (Lu and Liang 2012). It has been documented that vision is the primary source of information for equipment operators to avoid collisions with other entities on construction jobsites (Hirabayashi et al. 2006). The importance of clear, unobstructed vision coupled with the inherent poor visibility that operators of equipment such as dump trucks, loaders and excavators deal with due to blind spots (Teizer et al. 2010a) and other issues suggests that equipment monitoring and supplementary visual guidance can play a critical role in jobsite safety.

Operators in some tasks have the added cognitive burden of using judgment and estimation while performing their activities. Examples of this are excavation in presence of buried utilities, drilling in reinforced concrete slabs and tunnel boring operations that involve manipulation around occluded infrastructure objects such as underground utilities

and ductwork embedded in concrete walls. In the case of buried utilities, any errors on the part of an operator can rupture a gas line leading to an explosion, damage a water or sewer main leading to flooding of the jobsite or sever an electric conduit resulting in the risk of electrocution or loss of power to homes and businesses (Brammer 2006, New York Times 1995, Wilder 2010, WRAL 2007).

The type of equipment that can be monitored can vary from jobsite to jobsite, and even a single jobsite can have monitoring requirements that span different categories of equipment. The type of sensing mechanism used to record the position and orientation of equipment is also an operation- and equipment-dependant parameter. Thus, any monitoring framework intended for jobsite safety via supplementary visual guidance must be scalable and generic in order to be capable of monitoring equipment and operations across a broad range of jobsites and construction activities.

Equipment monitoring on a construction jobsite can be achieved using various approaches. Video surveillance, additional supervision, computer vision, Global Positioning System (GPS), and Radio Frequency Identification (RFID) are all examples of monitoring and control methods (Azar and McCabe 2012, Navon 2007). This chapter describes the limitations with these existing methods, and introduces a sensor-based real-time 3D visualization and geometric proximity monitoring method. Through this method, an equipment operator is provided a combination of visual assistance through real-time 3D visualization, and audio visual warnings through proximity monitoring, that together can help prevent any impending collision or accident. The following sections describe the

details of the proposed framework for providing real-time sensor updates by mapping field-based sensor data to corresponding equipment components in a 3D virtual world.

One construction operation in which equipment operators are routinely challenged on their skill and judgment, often in equal measure, is excavation, especially when carried out in the presence of underground utilities. Excavation is thus selected by the authors as an illustrative example of an operation where monitoring of equipment can provide additional assistance to operators to help them perform the activity safely. As the existing buried utilities are covered by dirt and soil, an operator cannot be certain of their location in relation to the equipment's end-effector (a bucket in the case of a backhoe or excavator). Excavation thus ends up being one of the leading causes of damage to underground utilities (Glink 2012, PHMSA 2012).

Whenever a buried utility is struck during excavation, it can result in loss of life, injuries, damage to property and a disruption to general life and commerce. The years from 1992 through 2011 saw a total of 346 major excavation-related accidents in the US that correspond to over 33% of all accidents involving buried pipelines. These accidents resulted in 148 fatalities, 532 injuries and over $98 million in property damage (PHMSA 2012). Excavation is thus an ideal operation to investigate the benefits of equipment monitoring for improving safety.

In the presented research, the authors instrument a backhoe loader with orientation sensors that monitor the rotation of the boom, stick, and bucket, and track the articulation

of the machine (and consequently the position of the bucket end-effector) in real-time. A generic and scalable framework for transmitting real world sensor data to update 3D equipment models inside a graphical virtual world for concurrent visualization is presented. The developed framework can be used to visualize any construction operation inside a dynamic 3D world as it occurs simply by outfitting the real equipment pieces with appropriate sensors and connecting them to their virtual counterparts.

5.2 Real-Time 3D Visualization

3D visualization can be of two categories – real-time and post-processed. In the case of post-processed visualization, the input is provided entirely from a simulation model (visual simulation), or from pre-recorded data of actual resource locations (trace simulation) (Kamat and Martinez 2001). However, in the case of real-time visualization, particularly when an ongoing operation is concurrently represented in a virtual world, there is an inherent need to link the real world to the virtual world. For real-time visualization, it is essential for the position and orientation sensor data from the jobsite to update the corresponding 3D equipment models in the virtual world in order to maintain a valid geometric display and related operational state.

The use of 3D visualization and geometric proximity monitoring has key advantages over other methods, especially when considering a construction jobsite. The use of conventional video cameras for monitoring safety on the jobsite is limited by physical constraints. Video cameras can limit the field of view in relation to the jobsite and equipment (Huber et al. 2009). Additionally, it may not be feasible to place a camera in a

location that provides the best perspective view of the equipment. Real-time 3D visualization when combined with geometric proximity monitoring provides spatial context in addition to visual guidance, i.e. the virtual environment can help "see" obstructions (e.g. buried utilities) that might not be visible either through video cameras or by human operators themselves. Real-time concurrent 3D visualization is thus a key requirement for effective equipment monitoring in any context aimed at improving jobsite safety, rather than mere observation or recording alone.

5.3 Literature Review

Equipment tracking and monitoring can occur at the macro- and micro-level. Macro-level tracking refers to those techniques where the location of the equipment in the global space is of interest to the site or fleet manager. Examples of this are fleet tracking applications for trucks, cars and other assets (Derekenaris et al. 2001, Sterzbach and Halang 1996, Zarazaga-Soria et al. 2001). At the macro-level, the user is primarily interested in the location but not the articulation (roll, pitch and yaw angles) or the details of sub-component orientations. At the macro-level, it is common for several resources to be tracked at the same time (Derekenaris et al. 2001, Sterzbach and Halang 1996, Zarazaga-Soria et al. 2001).

Micro-level tracking on the other hand is defined as that which occurs predominantly at a per-equipment level where the position of the equipment in global space along with detailed orientations of sub-components are monitored and displayed to the user for real-time visual analysis. It must be noted however that some information from the micro-

level monitoring can be used at the macro-level such as a site manager overseeing the operations of several equipment on an earthwork jobsite by viewing the position of all equipment on site.

The authors' proposed methodology is targeted at micro-level equipment monitoring and the remainder of this section describes existing micro-level tracking and monitoring approaches in civil and construction engineering. The Denavit-Hartenberg notation was used by Lu and Liang (2012) to develop a kinematic model for simulating the movement of a backhoe excavator, an example of articulated construction equipment. The use of ultra wideband (UWB) technology for tracking, monitoring, and estimating the pose of cranes was demonstrated by Zhang et al. (2012) by attaching UWB tags along the crane's boom and tip. Radio Frequency (RF) technology has been implemented to provide real-time warnings to RF-tagged equipment operators and workers by triggering alarms when the distance between pieces of equipment and/or workers drops below a safety threshold (Teizer et al. 2010b).

Lytle and Saidi (2007) of the National Institute of Standards and Technology developed a method to track the 3D position of a robotic crane using laser-based 3D site measurement system to provide position and orientation information to reduce errors in an encoder-based control system and also to map the crane's location relative to other components in the work environment. Sensor data and related CAD models were presented to the user in a visualization system (Lytle and Saidi 2007). Oloufa et al. (2002) have demonstrated the use of GPS technology for tracking equipment on a construction site and providing

warnings to operators about impending collisions between two moving equipment. The use of real-time kinematic GPS was tested and demonstrated by Roberts et al. (1999) for construction plant control and monitoring applications. Peyret et al. (2000) used a combination of GPS sensors to track the motion of an asphalt paving machine's blade by measuring its location in the vertical plane.

A combination of orientation (rotation) sensors, Global Positioning System (GPS) and laser technology has been used in applications for earthwork, grading and compaction operations to demonstrate equipment monitoring and control (Leica Geosystems 2012, Trimble 2012). The use of computer vision technology has been demonstrated for the recognition of construction equipment on earthmoving jobsites that has a potential to detect, track, and measure the productivity of stationary and mobile equipment (Azar and McCabe 2012). The field of equipment teleoperation is an example of micro-level monitoring where the operator is not present in the equipment and controls the equipment remotely. Steffen et al. (2007) demonstrated vehicle teleoperation through use of data from GPS and heading sensors by transmission over a wireless network and presentation through a 3D virtual display system.

The authors identify two primary limitations in the existing approaches: First, some of the existing approaches are limited by their narrow scope in being applicable to only a specific sensor-type and/or equipment type. Being applicable to any equipment type commonly found on a construction jobsite is a key requirement due to the number of different equipment pieces that may be present on any medium to large project. Second,

monitoring of equipment without providing operators with concurrent 3D visualization and proximity monitoring information limits its effectiveness due to the limited information being presented to the operators.

5.4 Overview of the Proposed Methodology

Equipment used on construction jobsites is very often articulated in nature i.e. its individual components are linked through joints that allow rotation about their pivot. Examples of these types of equipment are hydraulic excavators, backhoe loaders, graders, dump trucks and haulers. Monitoring such equipment involves recording the rotation and translation the equipment undergoes. The data collected is then transmitted to a 3D virtual world so that the real and virtual worlds can be geometrically correlated to one another. The 3D models in the virtual world when combined with the stream of position-orientation data can be used for carrying out proximity analysis and collision detection. Finally, the 3D visualization and related analytical output are presented to the operator. The entire process occurs in real-time or near real-time so that the output can be used by operators in their decision-making process.

Thus, it can be seen that being able to represent and simulate dynamic articulated equipment from the real world jobsite concurrently in a 3D virtual world requires a link joining the real and virtual aspects of the operation. There is an added level of complexity due to numerous types of construction equipment often being present on a jobsite, at any given instant. Similarly, an equipment's position and its sub-components' orientation can be recorded and monitored through a wide range of sensors and techniques. It follows

that any proposed methodology is required to be generic in its scope if intended to be applicable to a broad range of projects. The proposed methodology is represented schematically in Figure 5.1 with the data from the real world interfacing with individual components of the equipment in the virtual world.

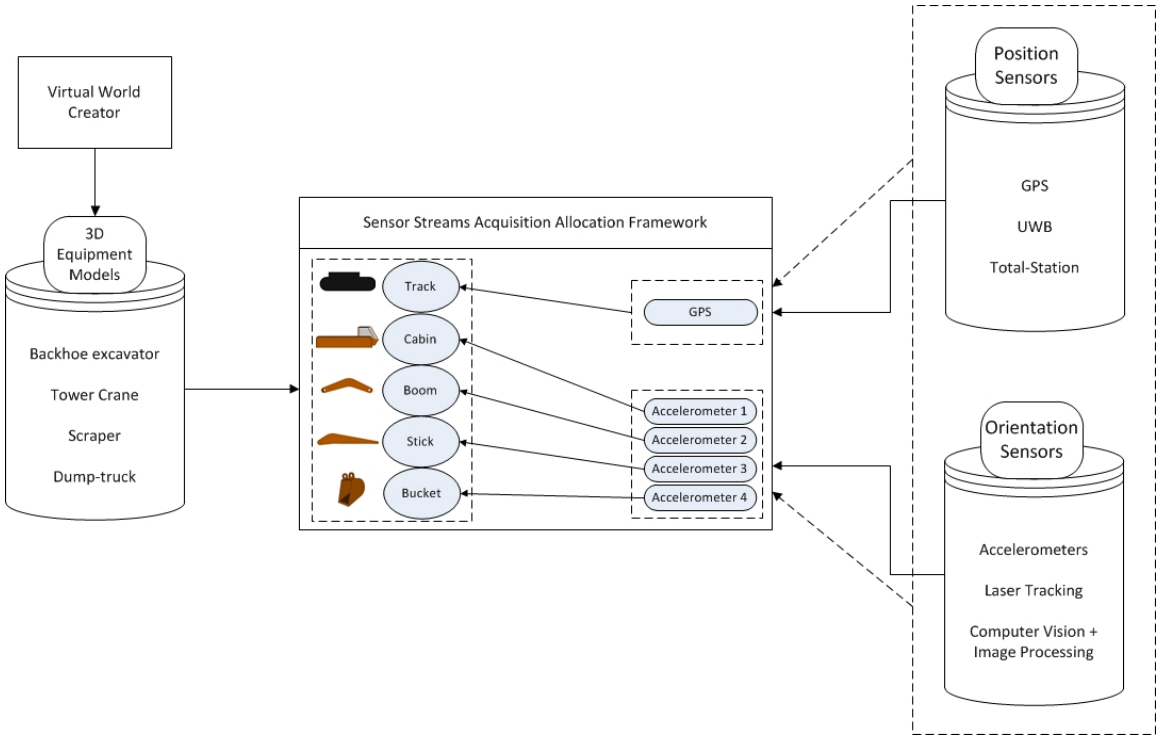

Fig. 5.1: Proposed methodology for creating a link between real world sensor data and 3D virtual equipment models

Figure 5.1 uses an excavator model to schematically show how the equipment's components can be connected to position and/or orientation sensor sources from the real world. A generic framework is capable of performing a similar allocation of sensor streams to different equipment such as a grader, crane, or backhoe loader based on the configuration of sensors installed on them. Similarly, the position and orientation sensors

can vary from one jobsite setup to another, examples of which are presented in Figure 5.1. Thus the ability to work with non-restrictive configurations of sensors and 3D equipment models is an essential part of the framework.

5.5 Technical Approach

In order to be able to replicate the real world motion of articulated equipment concurrently in the virtual world, understanding of the underlying equipment motion concepts is essential. This is primarily due to the complexity inherent in equipment such as backhoes and excavators, where it is not feasible to obtain the location of the end-effector in a direct manner. These limitations in turn are caused by potential damage the sensors can suffer if placed directly on the end-effector (bucket) where that can come in contact with dirt and soil over the course of operations. The remainder of this section describes the key technical challenges involved in concurrently replicating the translational and rotational motion of equipment inside a 3D virtual world.

5.5.1 Kinematics

Kinematics is the branch of mechanics that deals with the study of motion of a singular object or a group of objects without considering their causes (Beggs 1983). The field of kinematics itself is further divided into two approaches – forward and inverse kinematics. Forward kinematics refers to the use of the kinematic equations to compute the position of an object's end-effector from specified values for the object's joint parameters. Inverse kinematics on the other hand is the reverse procedure of forward kinematics. It is used to compute the angles and/or lengths of individual links and joints of the object and the

rotations and translations that the object must undergo in order to be present at a required (predetermined) final position and orientation.

In the case of equipment monitoring and operator assistance, the joint angle rotations are recorded by the sensors and thus the end-effector's position and orientation, $P_{end\text{-}effector}$ is computed through a forward kinematic process where $P_{end\text{-}effector}$ can be mathematically stated as 'f(Θ, L)' for every rotational joint and arm from the object's base to the end-effector, where 'Θ' is the rotational angle of the joint and 'L' is the length of the arm rotating about the pivot for every joint. This relationship is represented graphically in Figure 5.2 using the example of an excavator. The reader is referred to Garg and Kamat (2012) for additional details on the kinematic equations and process.

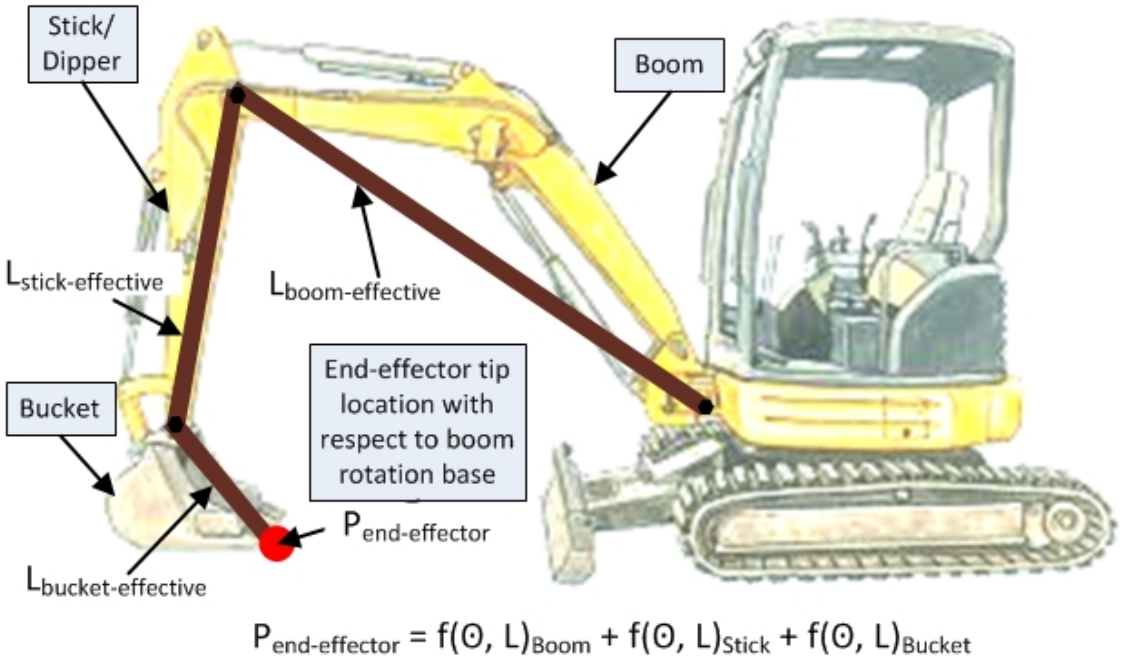

Fig. 5.2: Backhoe side view with schematic kinematic chain representing its boom, stick, and bucket articulation

As seen in Figure 5.2, the bucket's position and orientation depends upon the cabin's position and the sum of the lengths and the rotational angles of the boom, stick, and bucket respectively. The lengths of the boom, stick and bucket are considered from pivot-to-pivot so that the principles of forward kinematics can be applied to the computation. It must be noted that there are limitations on sensor placement upon equipment due to physical constraints and the harsh environment the equipment end-effector operates in. As a result, placing a position sensor on the end-effector that comes in contact with soil and other materials is difficult to achieve. Thus the end-effector's global position i.e. position of its tip in relation to other entities on the jobsite must be computed in an indirect manner.

It follows that determination of the global position of an end-effector tip has two computations associated with it. The first is the position aspect of the equipment on the earth's surface i.e. on the jobsite. The second involves the articulated chain of linked elements associated with the equipment. Thus, in the case of the backhoe shown in Figure 5.2, a GPS receiver is placed on the top of the backhoe to provide the equipment position on a jobsite in terms of latitude, longitude and altitude. In some cases, a pair of GPS receivers is placed on the cab of the equipment to provide more accurate position as well as orientation (rotation) of the cab (Roberts et al. 1999). Data from tilt measuring sensors placed along the boom, stick and bucket arms in combination with lengths of the respective components can provide the distance of the end-effector tip, measured from the articulated chain base i.e. the pivot point of the boom.

Based on the placement of the position sensor on the equipment, there may be an offset between the position reported by the sensor and the base of the articulated chain. Accurate integration of the two data computations requires the inclusion of the offset distance between the base of the articulated chain and the position reported by the position sensor. The combination of every f(Θ, L) along the articulated chain provides only the position of the end-effector with respect to the base of the articulation. In order to convert this value to its global position so that position value can be compared to other entities on the jobsite, the base pivot offset is required.

Every element along the kinematic chain has a set of rotation axes associated with it. The origin of the local axes corresponds to the rotation pivot of the element under consideration. At the same time, sensors placed on the equipment components record the rotation with respect to a set of global axes that are common for all the elements making up the equipment. Thus, there exist a set of global axes as show in Figure 5.3. These axes are called the X-, Y- and Z-Axes and are defined such that X-Axis points in the horizontal direction to the right, Y-Axis orthogonal to the plane defined by X- and Z-Axes and in a direction pointing away from the reader, while the Z-Axis points in the vertically upward direction. The three commonly used terms to describe rotation of a body in 3-dimensional space are roll, pitch and yaw. Roll is the rotation experienced about the X-Axis; pitch is the rotation about Y-Axis and yaw, the rotation about Z-Axis as shown in Figure 5.4.

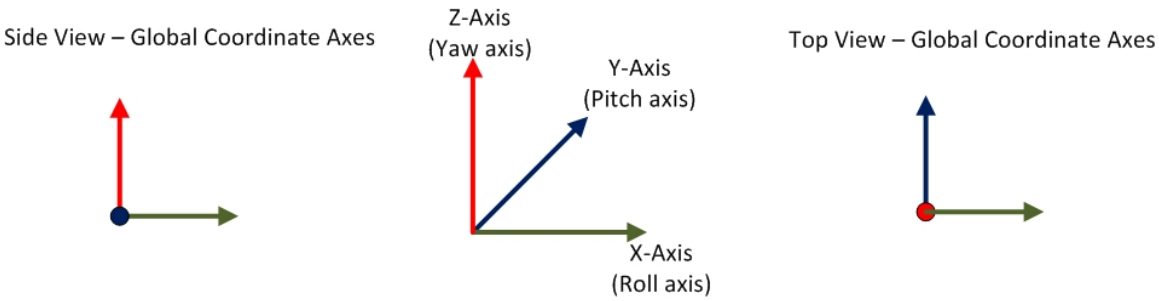

Fig. 5.3: Global coordinate axes – X, Y and Z about which a body experiences roll, pitch and yaw respectively

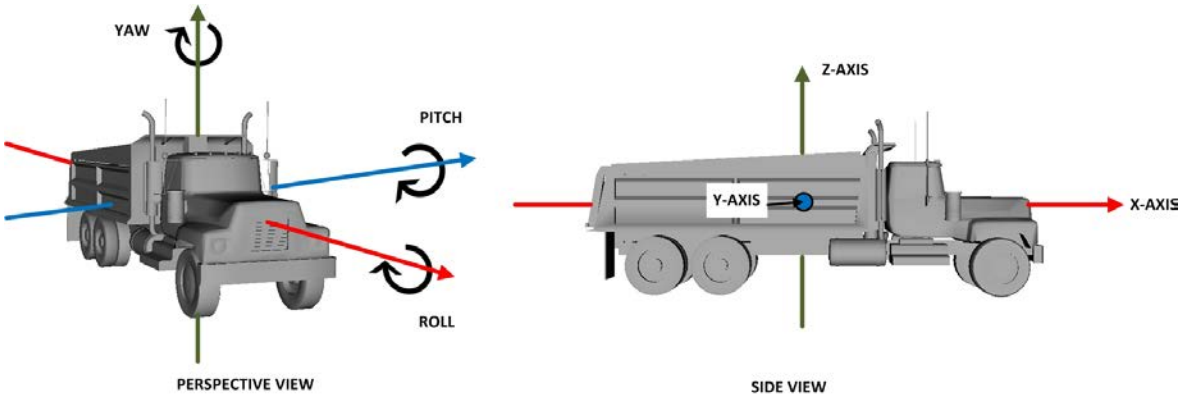

Fig. 5.4: Angles described in terms of roll, pitch, and yaw about the X-, Y-, and Z-Axis in both real and virtual worlds

Equipment components in an articulated chain have a parent-child hierarchical relationship. Hence, in the case of an excavator, the track component is the parent of the cabin, boom, stick and bucket elements. Similarly, the cabin component is the parent of the boom, stick and bucket components. In such a parent-child relationship, any rotation or translation experienced by the parent is implicitly transferred to the child entities. For example, anticlockwise rotation of the cabin by 90 degrees about the Z-Axis, results in rotation of the cabin as well as its child elements - boom, stick and bucket by the same

magnitude. However, the track component, which is the cabin component's parent element, is unaffected by this rotation. This behavior is represented in Figure 5.5.

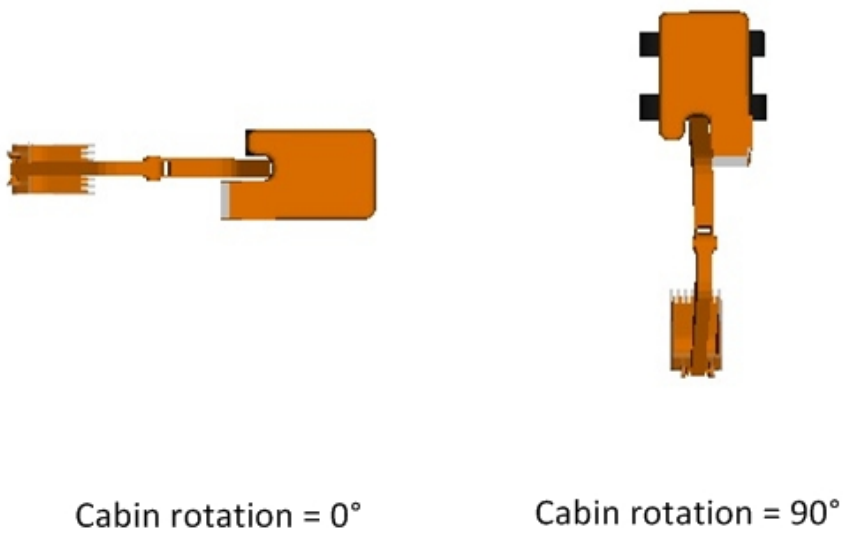

Fig. 5.5: Rotation of component (cabin) results in automatic rotation of child components (boom, stick and bucket) but not the parent component (tracks)

Due to such rotation and translation of individual components, the position and orientation of the local origin of the coordinate axes gets altered. This results in change in direction of local X, Y and Z axes. For example, rotation of the boom component in an anti-clockwise direction results in corresponding rotation of the stick and bucket components by the same magnitude. Due to this, the local axes direction of X, Y and Z differ from their global directions i.e. directions corresponding to zero translation and rotation. This is represented in Figure 5.6, through side and top views of an excavator.

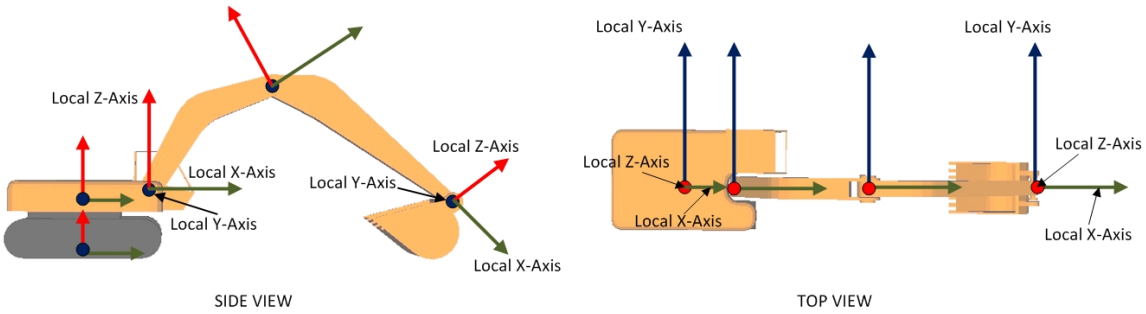

Fig. 5.6: Local rotation axes for equipment components

The cumulative effect of the parent components' rotation on a given element's local axes represents an important parameter in computations as the rotation angles reported by orientation sensors are typically computed with respect to absolute global axes (e.g., the horizontal and vertical directions are computed with respect to the ground surface). Thus rotations imparted to the 3D models representing the components in the real world need to be geometrically transformed so that the real world rotation from the sensors correlates to the existing rotation value of the 3D model component. The transformation is achieved by converting local rotations of individual components to their respective global rotation values.

5.5.2 Kinematic Equivalence

A 3D model used to symbolize equipment in the real world can represent the real world to varying levels of realism and detail. However, for 3D visualization and proximity analysis of the operation to be valid and useful to the operator, the authors have identified kinematic and dimensional equivalence as being essential requirements for a 3D model. Kinematic equivalence, in this context, refers to the property of a 3D model to concurrently mirror in the virtual world, the rotational and translation motion that a piece

of equipment and its sub-components undergo in the real world. This implies that the rotation or translation of a component resulting in a position change of an end-effector in the real world, must be replicated in the virtual world in such a manner that the end-effector has the identical global position in the virtual world as it has in the real world.

The effect of kinematic non-equivalence is most evident in the case of objects having curved bodies or objects having bends in their physical makeup such as booms and sticks/dippers. In such objects, the physical characteristics result in difference between the axis corresponding to a certain edge and the pivot-to-pivot rotation axis. For most equipment components, the sensors are often placed along an edge of the body, as shown in Figure 5.7, to ensure its position remains fixed during the course of operations and the sensors record the angle of a known edge in the real world.

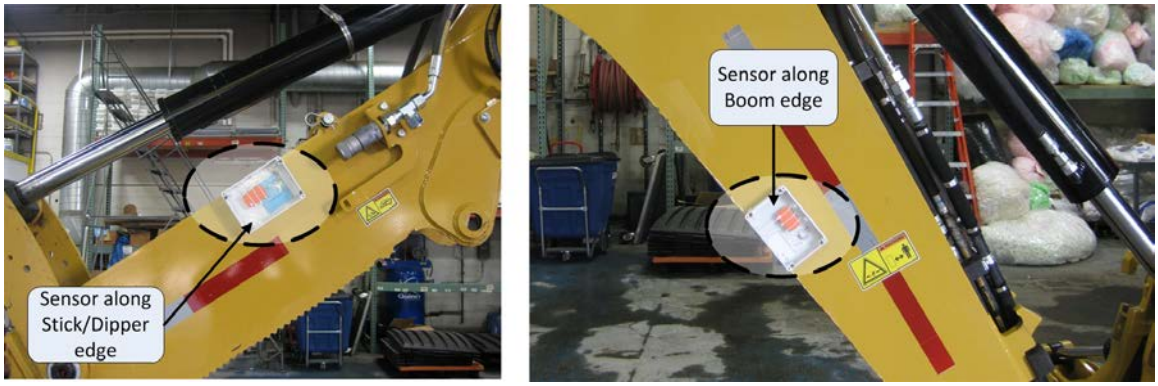

Fig. 5.7: Orientation sensors placed along the upper edge of the stick/dipper (left) and lower edge of the boom (right)

The problems associated with kinematic equivalency are graphically represented in Figure 5.8. In order to ensure that the 3D model in the virtual world has kinematic

equivalence to its real world counterpart, the angular offset between the actual rotation axis (represented by dashed line in Figure 5.8) and rotation axis corresponding to the sensor and boom edge (represented by solid line in Figure 5.8) needs to be accounted for while transmitting orientation values from the real to the virtual world. In the specific scenario of the boom object shown in Figure 5.8, the angular offset needs to be subtracted from the orientation sensor angle before being applied to the 3D virtual boom model. This ensures that the boom tip will have the same global position in the real and virtual worlds.

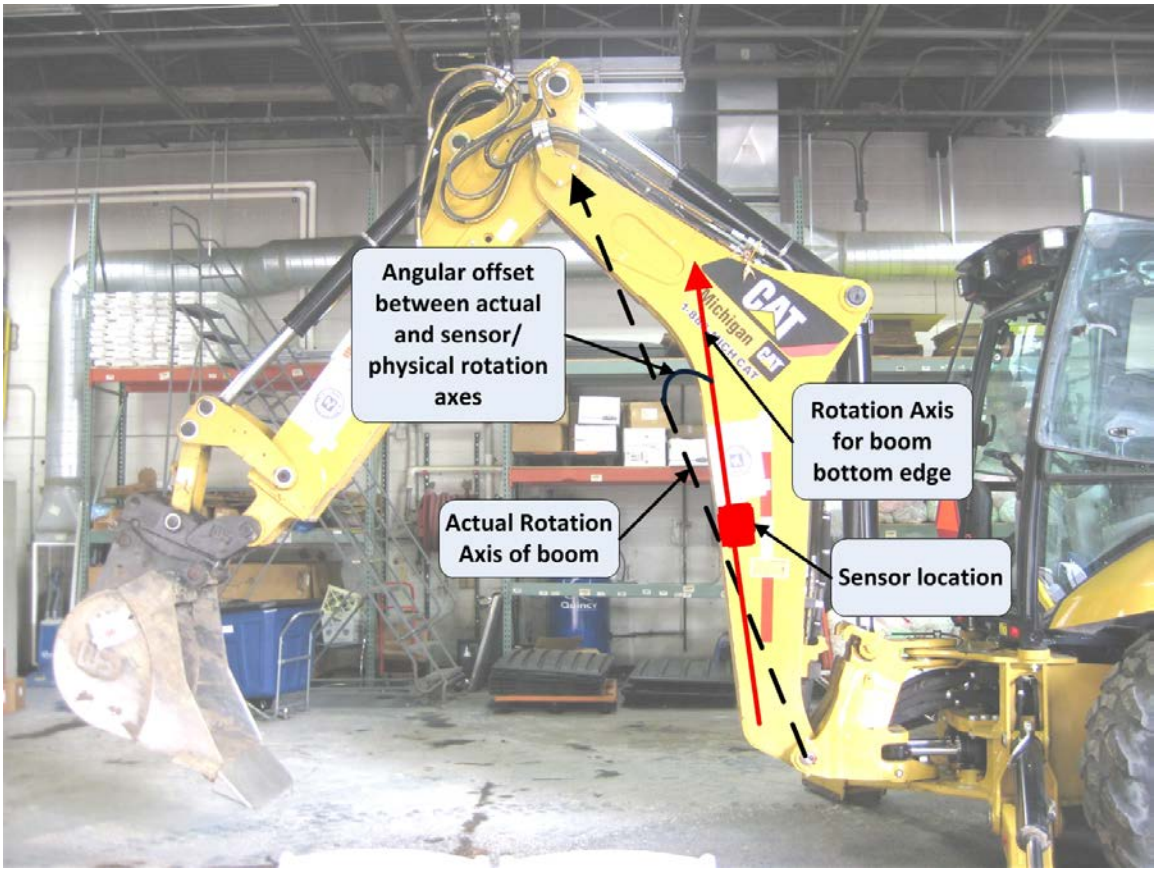

Fig. 5.8: Difference in actual rotation axis and rotation axis corresponding to physical edge of object (boom)

In addition to kinematic equivalence, the authors also introduce the concept of dimensional equivalence. This refers to the characteristic of a 3D model and its sub-components to represent the real world equipment's constituent components in size and placement. In some cases, when the 3D models used to represent equipment in the virtual world do not have identical dimensions to the real world equipment components, the following constraints are identified as being critical. First, any sub-component forming part of the overall 3D equipment model must have the identical dimensions to the real world equipment component such that the extents along X, Y, and Z axes (i.e. length, width and height dimensions) are identical in the real and virtual world.

Through this constraint, rotation of equipment components in the virtual world results in the position of the component extremity being identical to the real world. Without dimensional equivalence, use of correct sensor rotation values will not provide the same global position for the component's extremity and in turn that of the end-effector. Furthermore, for the geometric proximity analysis carried out on equipment components to provide values that are representative of the distances between the equipment and other jobsite entities (e.g., buried utilities), the dimensional extents need to be identical for the real and virtual world components.

The second constraint with respect to dimensional equivalence relates to location of base pivot joints and their height with respect to the ground surface. In the case of the kinematic chain representing the boom, stick and bucket, shown in Figure 5.2, the

position of the end-effector tip in global space is dependant upon the height of the joint corresponding to boom rotation pivot. Thus, in addition to accurate rotation angles and dimensions, the height of base pivot points and their distance with respect to other pivot joints is critical for accurate replication of the real world equipment operation inside a 3D virtual world.

5.5.3 Equipment representation in a concurrent Virtual World

As described in the previous sections, equipment monitoring is achieved through a combination of sensors placed on equipment and replication of the equipment movement (translation and/or rotation) in a 3D virtual world. This virtual world provides operators with visual assistance as well as additional spatial context information to allow them to perform their task better. Thus, it becomes evident that representation of equipment in 3D in an accurate and real world representative manner is a primary need for the proposed equipment monitoring approach.

Every object that is present in a 3D virtual world appears as a single cohesive entity. However, some objects are made up of one or more constituent components. Articulated construction equipment is an example of this. For instance, a construction crane consists of a cabin, boom, cable and hook; a backhoe similarly consists of a tracked or wheel base, a cabin, a front end loader, and a boom, stick and bucket at its opposite end. Thus it can be seen that equipment of such type consists of more than a solitary sub-component, each of which is capable of translation, and/or rotation. The components in turn are linked to each other through a parent-child hierarchy, where translation or rotation of a

parent component results in corresponding movement in a child component. This parent-child hierarchical representation is captured in a data representation structure called scene graphs through their layout and structure (Cunningham and Bailey 2001).

Other fields such as computer science uses graph structures commonly, particularly where hierarchies are required to be represented. The two types of graph data structures that exist are undirected and directed graphs. Scene graphs are one of the most widely used implementations of graphs, particularly in the field of computer graphics. Scene graphs are essentially directed acyclic graphs. They have a hierarchical tree structure for managing the object transformations such as rotation, translation and scaling, level of detail, field-of-view culling, state transformations, and animation (Kamat and Martinez 2002).

Scene graph implementations are typically based upon the lower level graphics Application Programming Interfaces (API) such as OpenGL and provide an additional layer of abstraction between the implementation and the underlying hardware. The authors refer the reader to Talmaki and Kamat (2012) for more details on the use of scene graphs for creating 3D articulated equipment models. Through the use of scene graphs, equipment components can be ensured to be dimensionally as well as kinematically equivalent.

5.5.4 Server-Client Approach

One of the main challenges in designing effective equipment monitoring is the ability to connect a wide range of position and orientation sensors from the jobsite to 3D virtual model components for any of the commonly used equipment in the real world. Thus scalability and non-specificity (or generality) are identified as the two key requirements for transmission and linking of real world sensor data to the virtual world. Scalability refers to the ability to connect any number of sensors from the real world to 3D virtual equipment components in the real world. Non-specificity entails that sensors from the real world can transmit data and be connected to any equipment in a 3D virtual world that is representing its real world counterpart.

The two key requirements are enabled through a server-client approach to equipment monitoring. The server refers to the part of the framework that is responsible for recording and transmitting real world sensor data in an appropriate data format to the client. The client, in this case, equates to the virtual world i.e. 3D visualization and proximity monitoring that would consume the real world sensor data. Insulation of the client from the sensors used on a jobsite ensures that the visualization and proximity monitoring are not restricted by the type of position or orientation sensors that may be implemented. In addition, this arrangement also allows multiple sensor data streams to be made available for the client.

The common denominators for data transfer are specified through data structures for position and orientation. The position aspect of equipment on any jobsite can be

accurately described through its latitude, longitude and altitude. This corresponds to the X, Y and Z locations of the equipment inside the 3D virtual world. The latitude, longitude and altitude can be specified in geographic (spherical) or projected coordinate system units such that the locations of all entities are specified in the same system.

Orientation of a 3D model in a virtual world is specified through rotation about a specific axis or a combination of axes. By selection of appropriate axes of rotation in the virtual world, the rotations in real and virtual worlds can be correlated to each other. For example, the axis of rotation of sensors placed on equipment in the real world can be made equivalent to the global axes of rotation in the virtual world as described in the previous sections. Thus the rotation values from the server are specified in terms of roll, pitch, and yaw, about the X-, Y- and Z-Axis respectively as shown in Figure 5.4.

Through the specification of orientation angles in terms of roll, pitch and yaw, the orientation angles from the real world can always be correlated to the virtual world regardless of the sensors used. For example, the use of linear encoders to measure the extended lengths of pistons on the server side of an equipment monitoring operation can be modified at the server-side by converting the lengths of cylinder extension to equivalent rotation angles for the boom or stick. In this manner, the same visualization and proximity monitoring-based client can be used with both direct orientation sensors such as accelerometers and indirect approaches such as linear encoders. Thus, insulation of the client visualization and proximity monitoring from the sensors used on the server side ensure scalability and non-specificity as shown in Figure 5.9.

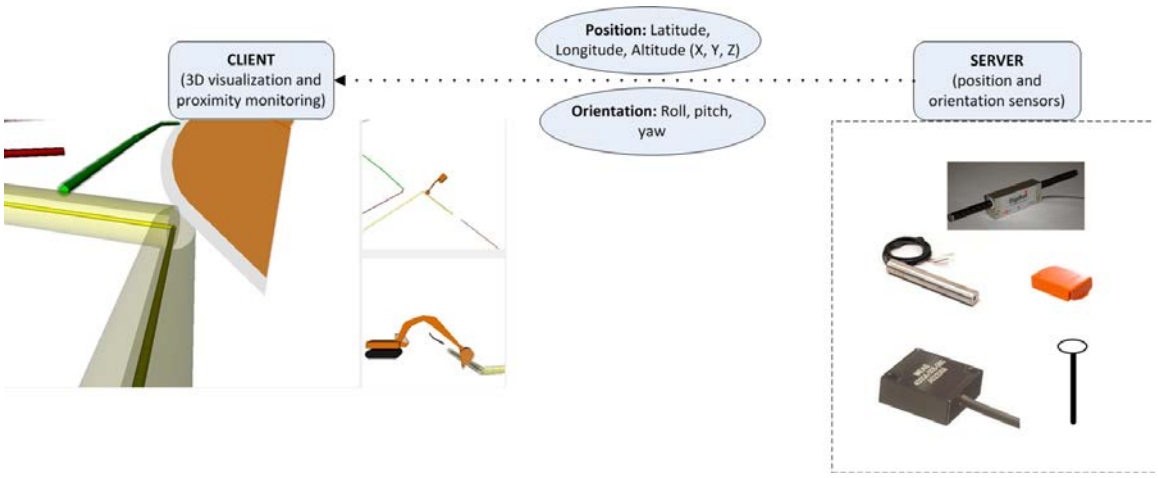

Fig. 5.9: Server-Client approach for transmitting position and orientation sensor data stream from the real to virtual world

5.6 Sensor Stream Acquisition Allocation (S2A2) Framework

In this section the authors introduce a framework developed to enable transmission of real world sensor data into a 3D virtual world. The framework is called the Sensor Stream Acquisition Allocation (S2A2) framework. The S2A2 framework is designed as an interface to an existing real-time 3D visualization system, SeePlusPlus that has been developed by the authors for improving safety in excavation operations through monitoring of excavators and providing visual guidance and warnings against impending collisions with buried utilities. The S2A2 framework is a link between articulated 3D equipment models present in the virtual world (SeePlusPlus) and sensor data streams from the real world. Figure 5.10 shows a schematic overview of the proposed equipment monitoring approach through real-time 3D visualization and proximity monitoring.

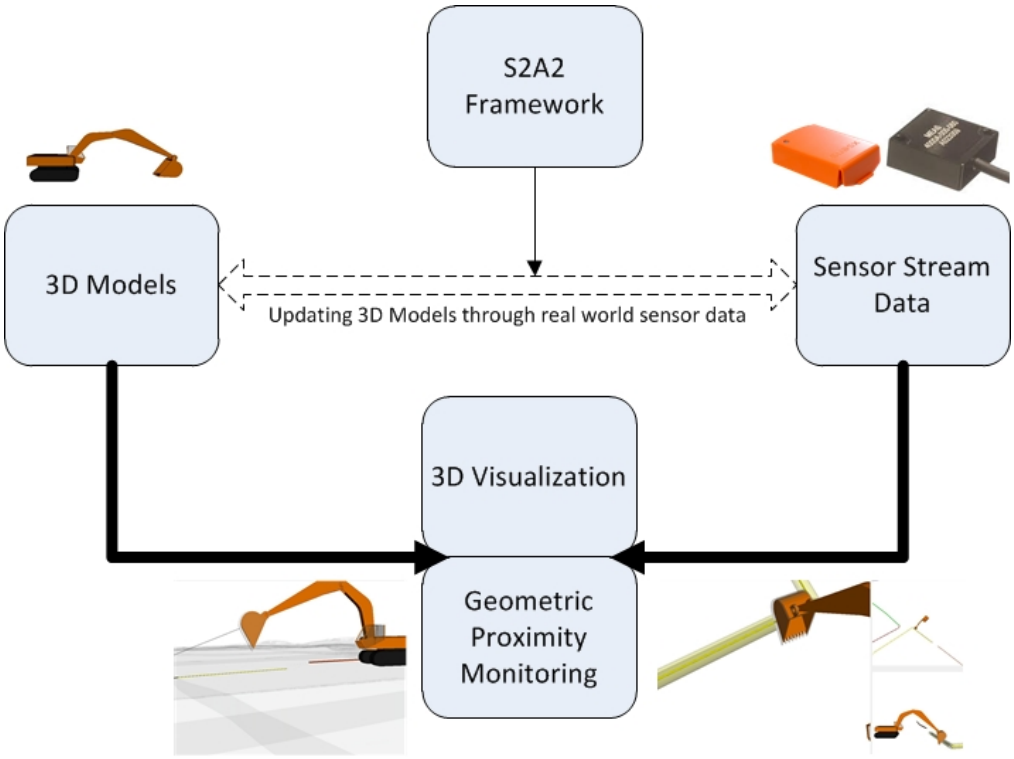

Fig. 5.10: Schematic representation of S2A2 framework for integration with real-time 3D visualization and geometric proximity monitoring

As seen in Figures 5.1 and 5.10, the process of connecting sensor data streams from the real world to 3D equipment residing in a virtual world depends upon the ability to expose to the user, the available 3D equipment components that can undergo rotation and/or translation. Furthermore, the 3D equipment must exhibit the same parent-child relationship in the virtual world as is evident in the real world. 3D equipment models used for such monitoring purposes are often built from individual sub-components rather than consisting of a single model that is incapable of movement in its sub-components. These requirements are implemented in another software tool designed for creating 3D equipment models to assist in monitoring operations. The tool is hereon referred to as the 'Virtual Equipment Builder' (VEB).

VEB is designed as a graphical interface that allows users to load individual equipment components as child (or parent) elements of other components and thus build up a complete equipment model in the process. The complete equipment model can be saved for use in the SeePlusPlus real-time 3D visualization system. VEB is designed and developed as an interactive tool that allows users to manipulate the view as well as change the position and/or orientation of components while creating the complete equipment model. A screenshot of VEB with a 3D excavator model created in it is shown in Figure 5.11.

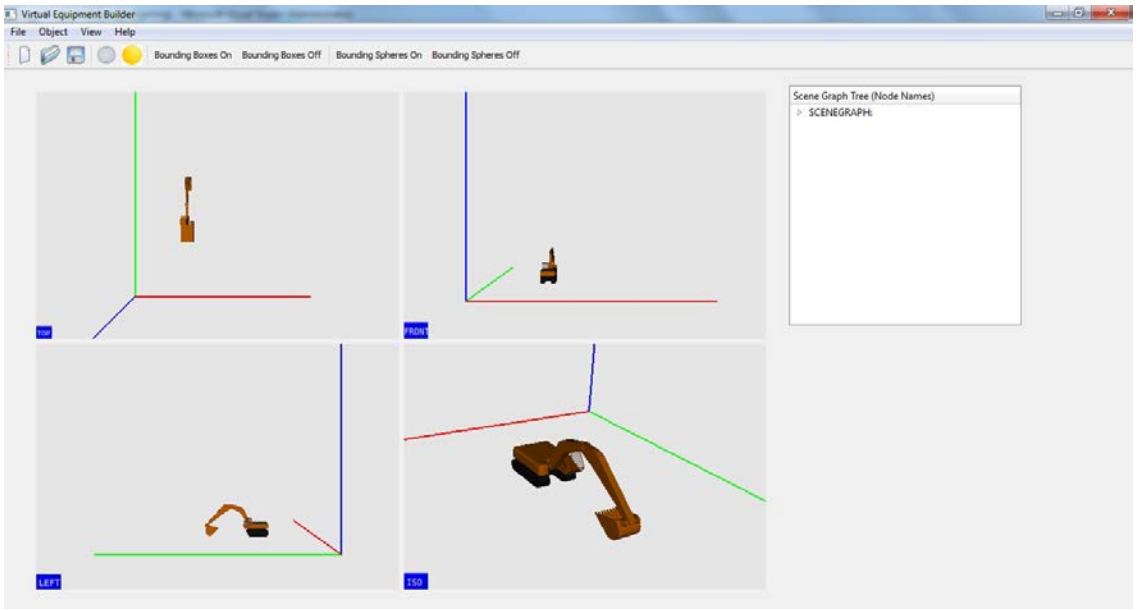

Fig. 5.11: Virtual Equipment Builder screenshot showing graphical interface and multiple views of a 3D excavator being built

SeePlusPlus is the 3D visualization system that provides users with visual guidance and also has an associated proximity monitoring and collision detection module. However, for the virtual world in SeePlusPlus to be representative of real world operations, a user

requires a method to connect real world sensor data to equipment components that make up the 3D equipment models. Figure 5.12 shows a screenshot of SeePlusPlus displaying a concurrent scene with an excavator and buried utilities. The terrain is rendered translucent to enable visualization of otherwise occluded utilities. Proximity monitoring in the scene is displayed to the user through three channels. First, numerical shortest distance to collision; second, lines drawn between bucket and utilities; and third, traffic signal colors-based warning showing green, amber and red for varying levels of safety and accident risk. Figure 5.12 also highlights the user command to activate the sensor connection interface for making the sensor data stream available from the S2A2 framework.

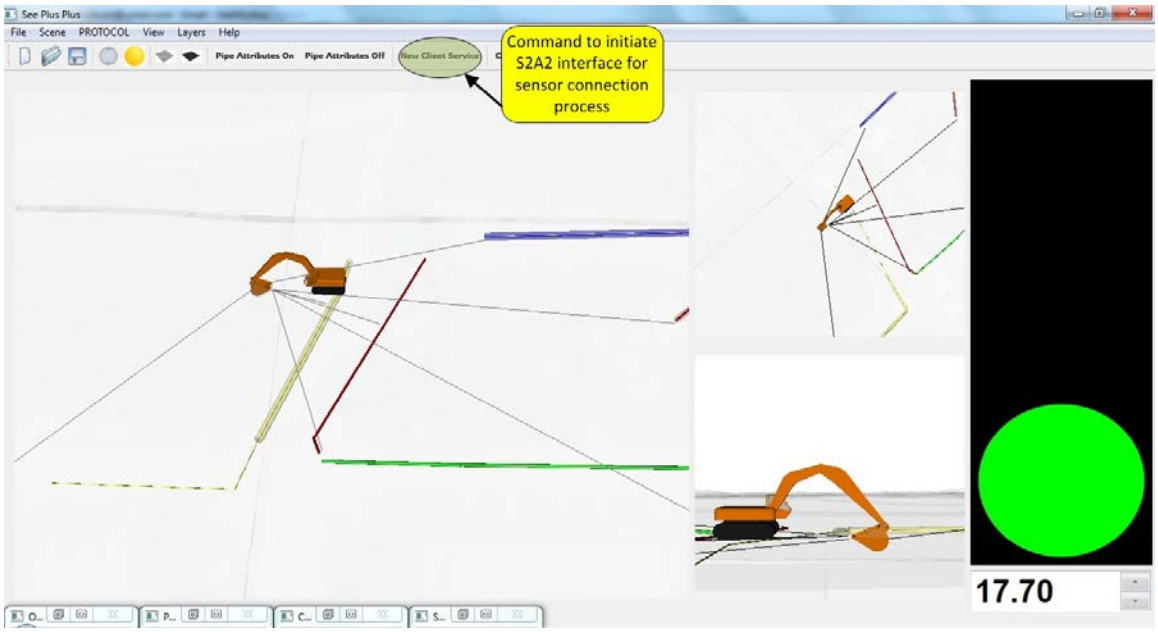

Fig. 5.12: Screenshot of SeePlusPlus showing a real-time scene containing an excavator and buried utilities and associated proximity monitoring between bucket and utilities, and S2A2 initiation command highlighted

The connection between server streams and the client is implemented through socket-based connections. In sensor implementations, data is transferred from the sensors to a server-side application wirelessly or through a physical connection. The server-side application then converts the raw data to position and/or orientation values that are required by the client application. Thus, the server- and client-side applications can run on the same physical machine and the use of socket-based connections ensures that data transfer is application-independent. The command to initiate a client-side connection to the server-side application provides the user with an interface as shown in Figure 5.13.

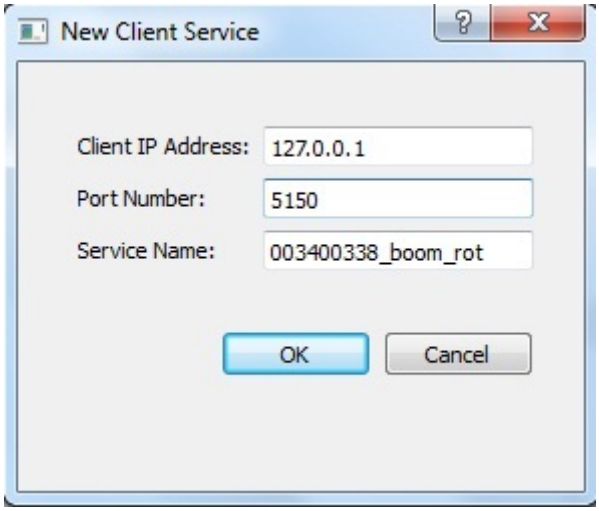

Fig. 5.13: Input dialog interface for creating new client service for connecting to a server-side data stream

A successful connection between client and server sides results in the new server stream being added to the list of available sensors in the S2A2 interface. The interface also presents the list of sub-components that make up the 3D equipment model. A screenshot of the S2A2 interface is shown in Figure 5.14. As is evident from the S2A2 acronym, the

interface also allocates sensor streams to user defined equipment components. The allocation is specified through a set of checkboxes that allows users to select what component of the sensor data stream may be used to update the selected equipment component. For example, selection of only Translate X, Translate Y and Translate Z options of a sensor stream ensures that only its position aspect would be used to update the selected equipment component.

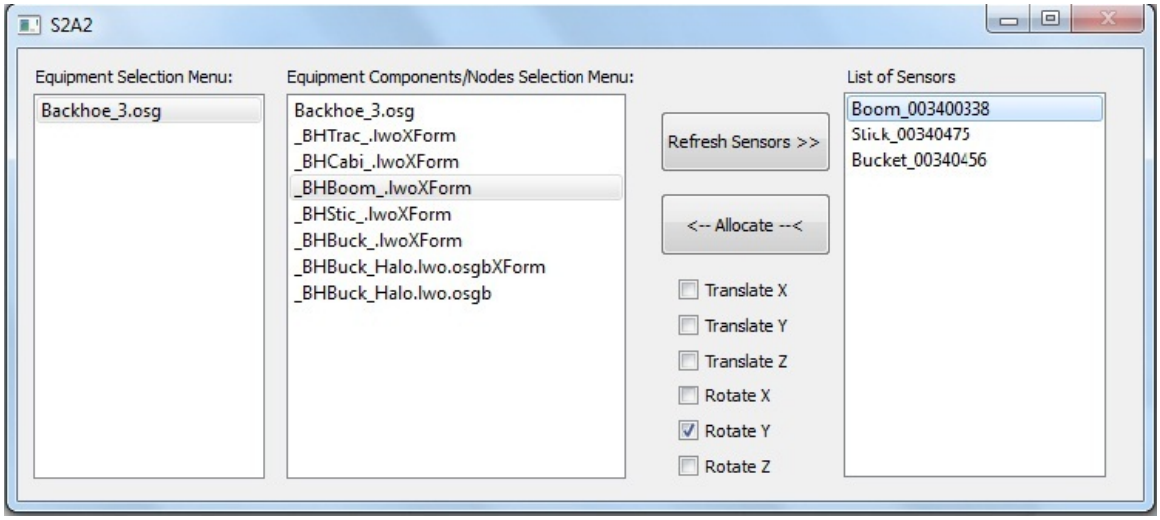

Fig. 5.14: Graphical interface for user-defined connections between real world sensors and virtual equipment components

In a similar manner, the rotation can also be specified by choosing one or more of roll (Rotate X), pitch (Rotate Y) and yaw (Rotate Z). Once an equipment component has been allocated a sensor stream, its position and/or orientation is updated in real-time as long as the sensor in the real world is active and transmitting data. Thus the S2A2 interface is designed to work together with the SeePlusPlus real-time 3D visualization application in providing real world sensor data to update the corresponding 3D equipment models. This is necessary to enable the monitoring of operations and improve safety.

5.7 Validation Experiments

In this section, the authors describe the procedure and setup used for carrying out validation experiments. The experiments are designed to demonstrate the functioning of the S2A2 framework and the accompanying 3D visualization when used to monitor and track a backhoe, and characterize the extent to which an arbitrary construction operation involving articulated equipment can be visualized in a real-time concurrent 3D virtual world. Through the S2A2 framework, changes in the equipment's articulation are transmitted to a 3D visualization in real-time. This position and orientation (pose) data is then used in the visualization to update the location and pose of corresponding dynamic entities. The updated 3D equipment components are then used in real-time proximity analysis to present distance and impending collision information to the operator of the equipment.

The backhoe used in these experiments was a Caterpillar 430 E IT (Caterpillar 2012). Screenshots from a simultaneous video recording of the real and virtual worlds is shown Figure 5.15. The articulation of the equipment's arm and end-effector was captured through a series of orientation sensors placed along its boom, stick and bucket. The orientation sensors used in the experiment were XSens MTw (XSens 2012). Bluetooth wireless technology was used to transfer pose data from individual sensors to the device running the 3D visualization. The accuracy of the sensors and their calibration is measured through the proximity monitoring framework that uses the pose updates as input.

Fig. 5.15: Images captured from a simultaneous video recording of the validation experiment showing backhoe in the real world and the 3D visualization

Through the 3D visualization and proximity analysis, the experiments demonstrate equipment monitoring in the following manner: In the first experiment, distances computed in the virtual world are compared to those in the real world. In the second experiment, the effect of audio and visual warnings on an operator's performance is investigated. The rest of this section describes the experiment details and results obtained.

5.7.1 Experiment 1

The focus of this experiment was on capturing and representing the real world equipment articulation in a 3D virtual environment. Hence, the position aspect of the equipment was discounted as the device was assumed to be stationary. Through the proximity monitoring framework, the distance between the equipment's end-effector (bucket in this case) and the ground surface on which the equipment was resting, was monitored at all times. If the equipment's articulation in the 3D virtual environment was an accurate representation of the real world, the distance between the end-effector and ground surface would be identical (after accounting for sensor-based inaccuracies) in both the real and virtual worlds.

During the test, the equipment's boom, stick and bucket was manipulated by the operator similar to regular operations. As the validation required distance values from both the virtual and real worlds, the operator was instructed to stop motion of the equipment's arm whenever a distance measurement was to be made in the real world. After the equipment had come to a complete halt, the distance between the end-effector (bucket) and the ground surface beneath it was measured using a measuring tape. This process was repeated for changeable configurations of the boom, stick and bucket. In total, 15 distance measurements were made in the real world. Corresponding distances displayed by the proximity monitoring framework in the virtual world were also recorded simultaneously. The values obtained from real and virtual world measurements are shown in Table 5.1.

Iteration No.	Distance in Real World (meters)	Distance in Virtual World (meters)	Real World – Virtual World (meters)
1	3.07	3.12	0.05
2	1.54	1.52	-0.02
3	1.14	1.16	0.02
4	0.63	0.66	0.03
5	0.00	0.17	0.17
6	0.66	0.68	0.02
7	1.62	1.62	0.00
8	2.18	2.08	-0.10
9	2.48	2.39	-0.09
10	2.84	2.71	-0.13
11	1.82	1.87	0.05
12	0.99	0.99	0.00
13	0.61	0.67	0.06
14	2.66	2.74	0.08
15	2.54	2.59	0.05

Table 5.1: Comparison of distance measurements made in the real and virtual world for varying configurations of boom, stick and bucket

5.7.2 Experiment 2

The second experiment was designed to test the latency of the S2A2 framework and effectiveness of the warnings provided by the 3D visualization and proximity monitoring

framework. The setup was similar to that in experiment 1, where the equipment was assumed to be stationary and only orientation coordinates were used in the analysis. Unlike in experiment 1, a tolerance check was introduced to warn the operator as soon as a preset safety threshold was breached. For the purpose of the test, distances of 2.0, 2.5 and 3.0 m were set as the safety threshold level between the end-effector and the ground surface. The proximity monitoring framework in this experiment was thus designed to warn the operator when the end-effector comes within the preset safety threshold distance to the ground surface.

Once again the test was carried out through multiple iterations with changeable articulation of the boom, stick and bucket. The operator was instructed to halt the motion of the equipment as soon as an audio-visual warning was provided. Once the equipment was stopped, the distance between the end-effector and the ground surface was recorded. This process was repeated ten times and results from each iteration along with the distance by which the operator had penetrated the preset buffers (2.0, 2.5, 3.0 m) are presented in Table 5.2. The significance of the results and their interpretation by the authors is presented in the following section.

Iteration No.	Buffer Depth (meters)	Distance from the ground surface (meters)	Penetration into safety buffer (meters)	Operating Speed (L = Low, H = High, V = Very High)
1	2.00	1.87	0.13	L
2	2.00	1.01	0.99	H
3	2.00	0.67	1.33	V
4	3.00	2.74	0.26	L
5	3.00	2.59	0.41	H
6	2.50	2.22	0.28	L
7	2.50	1.79	1.11	H
8	2.50	0.80	1.70	V
9	2.50	1.39	1.11	H
10	2.50	1.54	0.96	H
11	2.50	1.98	0.52	L
12	2.50	2.12	0.38	L
13	2.50	1.94	0.56	L
14	2.50	2.07	0.43	L
15	2.50	1.62	0.88	H

Table 5.2: Penetration depth of safety buffer for varying operating speeds and safety buffer depths (L = Low speed, H = High Speed, V = Very High Speed)

5.8 Discussion of Results

In experiment 1, the difference between end-effector to ground surface distance in real and virtual worlds can be attributed to the following factors. The first and most significant factor relates to the ground surface slope. The experiment was carried out in an indoor facility owned by the University of Michigan. The floor of this facility had a slight slope/gradient to it designed to allow water drainage. However, in the virtual world, ground surface was modeled as a perfectly flat surface due to non availability of the indoor elevation data for the experiment site. Thus, a positive or negative slope for the ground surface in the real world would result in vertical distance values (from bucket teeth to ground surface) that are less or greater than the corresponding values measured to a flat surface in the virtual world.

Second is the geometric difference between the bucket in the real and virtual worlds. The bucket of the backhoe was modeled from a pre-existing 3D model and thus resulted in unintended dimensional variances between the bucket teeth in the real and virtual world objects. The distance in experiment 1 was measured as the vertical separation distance between the bucket teeth and ground surface. Hence, dimensional equivalence is stated as a key requirement for accurate monitoring of equipment on jobsites.

The third source of error is based on the accuracy of angles measured and reported by the orientation sensors used in the experiment. The static accuracy of the MTw sensors used in the experiment is stated as being 0.5 degrees (XSens 2012). Thus, for a boom length of 2.75 m as in the case of the Caterpillar 430 E IT, a 0.5 degrees error results in a vertical

deviation of 0.09 m. The combination of these factors is attributed to the difference between real and virtual world values for distance between the backhoe bucket and ground surface, observed in experiment 1. The mean separation distance error with the standard deviation (shown in Table 5.1) is 0.06 ± 0.05 m.

Experiment 2 was designed to test the effect of audio and visual warnings on an operator when a preset safety threshold was breached by the end-effector. The experiment used safety thresholds of 2.0, 2.5 and 3.0 m. For each of the safety thresholds, the operator manipulated the equipment at varying speeds, categorized in Table 5.2 as ‘low’, ‘high’, and ‘very high’. Low corresponds to the angular speed when a bucket would be in close proximity to a surrounding object or when working in a narrow or confined space. High and very high correspond to the typical angular speed of the bucket that would be observed during the course of regular excavation operations. The safety threshold value was changed through the course of the experiment to ensure that the operator was responding to the warnings and not the learning-effect of the same threshold value. Different depth values also provided the opportunity to analyze the effect of larger safety buffers over smaller values.

The goal of the safety buffer is to provide a warning to the operator with sufficient reaction time to bring the equipment to a halt before the end-effector can come in physical contact with an object that may be occluded such as buried utilities covered by the soil. The effectiveness of a safety buffer is analyzed by computing the magnitude by which the buffer was breached by the bucket. Table 5.3 shows the list of values by which

the bucket breached the preset safety threshold (expressed as a percentage of the buffer depth) after the operator was provided with warnings. In Table 5.3, it can be observed that the magnitude by which the safety buffer was breached did not vary significantly when the safety buffer depth was changed between 2.0, 2.5 and 3.0 m. For example, at very high operating speed, the safety buffer was breached by 67% for a 2.0 m buffer and 68% for a 3.0 m.

Buffer Depth (meters)	**Operating Speed (L = Low, H = High, V = Very High)**	**% Safety Buffer Breached**
2	L	7
2	H	50
2	V	67
3	L	9
3	H	14
2.5	L	11
2.5	H	44
2.5	V	68
2.5	H	44
2.5	H	38
2.5	L	21
2.5	L	15
2.5	L	22
2.5	L	17
2.5	H	35

Table 5.3: Breach of safety buffer for varying operating speeds expressed as percentage of the buffer depth (rounded to nearest whole number)

The average percentage safety buffer penetration for L, H, and V operating speeds was found to be 15%, 38%, and 68% respectively. Thus it can be seen that the safety buffer depth depends on the operating speed of the equipment and the specific operating style of a given operator. The need for a dynamic safety buffer that adjusts itself based on the operating characteristics of a given operator is identified as one of the future goals of this research. In this way, the safety buffer provided gives the operator adequate time to take evasive action once a warning is sounded. During experiment 2, it was observed that the operator would often react late to the audio and visual warnings due to lack of previous experience with such a system. Thus, operator training with the associated monitoring systems in a simulator or actual machine is identified as an associated requirement for maximum benefit from such monitoring implementations.

5.9 Conclusion and Future Work

In this chapter, the authors investigated the types of equipment monitoring that occurs on construction and mining jobsites as well as in other commercial settings. The need for detailed or micro-level equipment monitoring was presented and the areas where its use can help reduce accidents and improve overall safety were described. The authors also presented a framework for equipment monitoring based on concurrent 3D visualization and real-time proximity monitoring using sensor-based input for updating 3D equipment components. The principles developed in the proposed methodology and technical approach were demonstrated through an interface for mapping sensor data streams to specific equipment elements. This interface was presented in context of a real-time 3D

visualization application for assisting excavator operators in preventing unintended strikes with underground utilities.

The chapter also describes two validation experiments to demonstrate the ability to simulate the real world motion of equipment in a concurrent 3D virtual world. The experiment simulates the motion of a backhoe loader's articulated arm through orientation sensors installed on its boom, stick/dipper and bucket. The results comparing accuracies in the real and virtual worlds are presented. In addition, the effect of a warning mechanism based on preset safety thresholds was investigated on the performance of the operator.

In the previous section, the authors introduced the concept of dynamic tolerance zones based on the speed of equipment operation, which in turn depends upon an operator. It can be seen from the results in Experiment 2 that higher operating speeds require larger tolerance zones. However, the ability to dynamically modify the tolerance zone size as a function of operator speed is identified as a future direction of this research. The future goals of this research include experimenting with tracking sensors such as linear encoders and other sensors that are pre-installed or retrofitted on equipment (Landberg 2002). The authors are also investigating the applicability of the proposed framework on other equipment types such as cranes, dump trucks and graders.

5.10 Acknowledgments

The presented research was funded by the US National Science Foundation (NSF) via Grants CMMI-927475 and CMMI-1160937. The writers gratefully acknowledge NSF's support. The writers also thank Mr. Jerome Schulte, Mr. Samuel Moran, and backhoe-operator Mr. William Sodt for their assistance in providing the equipment for carrying out the validation experiments. In addition, the writers would also like to thank Mr. Sean O'Connor for his assistance in sensor installation-related activities. Any opinions, findings, conclusions, and recommendations expressed in this chapter are those of the authors and do not necessarily reflect the views of the NSF, University of Michigan or the individuals mentioned herein.

5.11 References

Azar E. R., McCabe, B. (2012). “Part based model and spatial–temporal reasoning to recognize hydraulic excavators in construction images and videos”, Automation in Construction, Volume 24, July 2012, Pages 194-202, ISSN 0926-5805, 10.1016/j.autcon.2012.03.003.

Beggs, J. S. (1983). “Kinematics”, Taylor & Francis, p1.

Brammer, B. (2006). “Dubuque Company Pays $5,000 for Violating ‘One Call’ Law”, Iowa Department of Justice, Office of the Attorney General, <http://www.iowaattorneygeneral.gov/protecting_environment/2006/dubuque_company.html> (10/14/2012).

Castro-Lacoutere, D., Irizarry, J., and Arboleda, C.A. (2007) “Ultra wideband positioning system and method for safety improvement in building construction sites”, American Society of Civil Engineers Construction Research Congress, 2007, Grand Bahama Island, The Bahamas, May 2007.

Caterpillar (2012). “Caterpillar 430E/430 EIT Backhoe Loader”, <http://www.cat.com/cda/layout?m=308397&x=7> (10/14/2012).

Cunningham, S., and Bailey, M. J. (2001) “Lessons from Scene Graphs: Using Scene Graphs to Teach Hierarchical Modeling,” Computers &Graphics, 2001, number 4.

Derekenaris, G., Garofalakis, J., Makris, C., Prentzas, J., Sioutas, S., Tsakalidis, A. (2001). "Integrating GIS, GPS and GSM technologies for the effective management of ambulances", Computers, Environment and Urban Systems, Volume 25, Issue 3, 1 May 2001, Pages 267-278.

Garg, A., Kamat V.R. (2012). "Virtual Prototyping for Robotic Fabrication of Rebar Cages in Manufactured Concrete Construction" Proceedings of the 2012 CONVR Conference, Taipei, Taiwan.

Glink, I. (2012). "Backyard digging poses unseen hazards", CBS News <http://www.cbsnews.com/8301-505145_162-57507838/backyard-digging-poses-unseen-hazards/> (10/14/2012)

Hirabayashi, T., Akizono, J., Yamamoto, T., Sakai, H., and Yano, H. (2006). “Teleoperation of construction machines with haptic information for underwater applications”, Automation in Construction, 15(5), 563-570.

Huber, D., Herman, H., Kelly, A., Rander, P., and Warner, R. (2009). “Real-time Photorealistic Visualization of 3D Environments for Enhanced Teleoperation of Vehicles”, Proceedings of the 2nd International Conference on 3D Digital Imaging and Modeling, Kyoto, Japan.

Kamat, V.R., Martinez, J.C. (2001). “Visualizing simulated construction operations in 3D”, Journal of Computing in Civil Engineering, 15 (4) (2001) 329–337.

Kamat, V. R., and Martinez, J. C. (2002). “Scene Graph and Frame Update Algorithms for Smooth and Scalable 3D Visualization of Simulated Construction Operations”, Journal of Computer-Aided Civil and Infrastructure Engineering, Vol. 17, No. 4, Blackwell Publishers, Malden, MA, 228-245.

Landberg, L. (2002). "Lasers/GPS enhance machine control", <http://www.constructionequipment.com/lasersgps-enhance-machine-control> (10/14/2012).

Leica Geosystems (2012). “Leica iCON grade 42 - Intelligent Grading Systems” <http://www.leica-geosystems.com/en/Leica-iCON-grade-42_70038.htm> (10/04/2012).

Lu, M. and Liang, X. (2012). “Real-Time 3D Positioning and Visualization of Articulated Construction Equipment”, Computing in Civil Engineering (2012). June 2012, 196-203.

Lytle, A. M., and Saidi, K. S. (2007). “NIST research in autonomous construction.”, Autonomous Robots, 22(3), 211-221.

Mine Safety and Health Administration (2012). “Mining Equipment Camera Installation Tips for Best Results”, MSHA Accident Prevention Program, <http://www.msha.gov/Accident_Prevention/newtechnologies/initiatives/cameras/installtips.htm> (10/14/2012).

Navon, R. (2007). “Research in automated measurement of project performance indicators”, Automation in Construction, Volume 16, Issue 2, March 2007, Pages 176-188.

New York Times (1995) “Newark airport is closed as crew cuts power lines”, <http://www.nytimes.com/1995/01/10/nyregion/newark-airport-is-closed-as-crew-cuts-power-lines.html?pagewanted=2> (06/17/2011).

Oloufa A., Ikeda, M., and Hiroshi O. (2002). “GPS-Based wireless collision detection of construction equipment”, International Symposium on Automation and Robotics in Construction, 19th (ISARC). Proceedings, National Institute of Standards and Technology, Gaithersburg, Maryland. September 23-25, 2002, pp.461- 466.

Peyret, F., Betaille, D., and Hintzy, G., (2000). “High-Precision Application of GPS in the Field of Real-Time Equipment Positioning.*Automation in Construction*, 9: 299-314.

PHMSA (2012), “Pipeline Incidents and Mileage Reports – Serious Incidents” < http://primis.phmsa.dot.gov/comm/reports/safety/PSI.html > (10/14/2012).

Roberts, G. W., Dodson, A. H., and Ashkenazi, V., (1999). “Global Positioning System Aided Autonomous Construction Plant Control and Guidance”. Automation in Construction, 8: 589-595.

Steffen, M.A., , Will, J.D., Murakami, N. (2007). "Use of Virtual Reality for Teleoperation of Autonomous Vehicles", American Society of Agricultural and Biological Engineers Biological Sensorics Conference, Summer 2007, Available Online <http://gem.valpo.edu/~svl/research/pubs/useofvertualreality.pdf> (10/04/2012).

Sterzbach, B., Halang, W.A. (1996). “A mobile vehicle on-board computing and communication system”, Computers & Graphics, Volume 20, Issue 5, September–October 1996, Pages 659-667.

Talmaki, S. A., and Kamat, V. R. (2012). “Real-Time Hybrid Virtuality for Prevention of Excavation Related Utility Strikes”, Journal of Computing in Civil Engineering, American Society of Civil Engineers, Reston, VA. (In Review).

Teizer, J., Allread, B.S., Mantripragada, U. (2010a). “Automating the blind spot measurement of construction equipment”, Automation in Construction 19 (4), 491–501.

Teizer, J., Allread, B.S., Fullerton, C.E., Hinze, J., (2010b). Autonomous pro-active realtime construction worker and equipment operator proximity safety alert system. Automation in Construction 19 (5), 630–640.

Trimble GCS900. (2012). “Grade Control System - The Connected Machine” <http://www.trimble.com/construction/heavy-civil/machine-control/grade-control/> (10/04/2012).

Wilder, F. (2010) “The Fire Down Below”, Texas Observer, December 2, 2010 <http://www.texasobserver.org/cover-story/the-fire-down-below> (10/31/2011).

WRAL archives (2007), “Who's to Blame for Cary Gas Line Rupture?” <http://www.wral.com/news/local/story/1916911/> (06/17/2011).

XSens (2012), “XSens MTw - Wireless Motion Tracker” <http://www.xsens.com/images/stories/products/PDF_Brochures/mtw%20leaflet.pdf > (10/14/2012)

Zarazaga-Soria, F.J., Álvarez, P.J., Bañares, J.A., Nogueras, J., Valiño, J., Muro-Medrano, P.R. (2001). "Examples of vehicle location systems using CORBA-based distributed real-time GPS data and services", Computers, Environment and Urban Systems, Volume 25, Issue 3, 1 May 2001, Pages 293-305.

Zhang, C., Hammad, A., and Rodriguez, S. (2012). “Crane Pose Estimation Using UWB Real-Time Location System”, Journal of Computing in Civil Engineering 2012 26:5, 625-637.

Chapter 6

Conclusion

6.1 Introduction

This research successfully investigated and demonstrated the ability to represent and simulate real world operations concurrently in a real-time 3D virtual world. The research used excavation in presence of underground utilities as an example of an operation where such 3D visualization-based monitoring would benefit safety and productivity of an operation. Through the various facets of the research, the benefits of monitoring an operation through a virtual environment over conventional or existing approaches were presented. The need for monitoring operations and equipment on a construction jobsite stems from the unique nature of the challenges posed by the Civil and Construction engineering work domains.

6.2 Significance of the Conducted Research

Construction workspaces are unique and differ from those observed in manufacturing and other domains in the following ways: First, the jobsite is unstructured and dynamic (Son et al. 2008). Work stations in controlled environments such as manufacturing remain relatively unchanged over time. They are also designed to provide optimum visibility and spatial awareness to the worker. Construction jobsites, on the other hand often have physical constraints that are dictated by the available space thus limiting clear visibility for equipment operators and equipment.

Second, equipment and workers are present in close proximity (Teizer et al. 2010, Cheng and Teizer 2011). While manufacturing workplaces are often equipped with robots, their anchored base and constrained motion in a work station differs significantly from the motion of construction equipment that are relatively unconstrained and depend on the operator's control for motion on the jobsite. Third, the jobsite has workers belonging to different trades or crafts, often with no single level of command. Fourth, the work on some jobsites involves interaction with infrastructure that is buried or concealed and thus not visible to equipment operators or workers directly.

In the specific case of excavation operations involving buried utilities, excavator operators are faced with an additional set of challenges. It is currently required by law to have any proposed excavation pre-marked to show the approximate locations of buried utilities (Miss Dig Systems 2007). However, the surface markings are temporary in nature and can no longer guide the operator after the top surface has been removed. Any break in the excavation activity after the markings are removed results in the operator relying upon memory to recall the marked locations.

In addition, earth, dirt and/or soil covering the buried utilities and the lack of depth information of the utilities (Miss Dig Systems 2007) makes it difficult for the operator to judge the distance of the equipment's end-effector (i.e. the bucket) to the vicinal utilities in both the vertical and horizontal contexts. A common practice to avoid accidents is the use of hand-digging (Miss Dig Systems 2007). However, this process is time and labor-

intensive resulting in loss of productivity. Improving the operators' spatial awareness can reduce the volume of earth that is hand-excavated. Thus, excavation operators require additional visibility and spatial guidance to assist them for carrying out their task safely and productively. Excavation is an inherent activity in almost all construction projects. By addressing issues of excavation safety and productivity, this research has the potential to have a significant impact across the 1.2 trillion dollar US construction industry.

6.3 Research Contributions

It follows from the discussion in earlier chapters that a method to monitor the interacting entities on construction jobsites is critical for improved spatial awareness and safety. In particular, there is a need to provide visual assistance due to physically constrained jobsites and collision prevention warnings due to crowded work spaces around equipment having poor visibility and blind spots (Allread and Teizer 2010). The research breaks down interactions between entities on a jobsite into static-to-dynamic, dynamic-to-dynamic, and static-to-static. The frameworks presented in this research can monitor all these three types of relations. However, the focus is predominantly on the first two types as interactions between two static entities more commonly occurs during the pre-construction phase (Akinci and Fischer 2000).

The outcome of this research is a 3D visualization-based real-time monitoring framework that is designed to improve safety and productivity for excavation-related operations. The key contributions of the research are as follows:

- An analysis of the life-cycle that buried utility data passes through and subsequent division into the stages of Collection, Archival, Updating, and Visualization. Identification of limitations and improvement areas in existing practices for the four stages. The IDEAL framework for visualization of buried utilities was created by incorporating the suggested improvements.
- A 3D modeling system demonstrating buried utility data flow from GPS values collected in the field to creation of georeferenced (i.e. location-ware), color coded, attribute rich models for use in visualization and proximity monitoring applications. A framework to represent a real-world operation in a 3D virtual world through sensor-updated dynamic entities and georeferenced static entities, while maintaining an adequate level of abstraction for effective representation was also created.
- Algorithms for creating real-time proximity monitoring, tolerance checking and collision detection queries between entities present in a 3D virtual scene supplementing visual guidance by providing spatial information to operators.
- Methods for creating 3D articulated equipment models from individual components to represent real world equipment by ensuring kinematic equivalency
- A principle called Hybrid Virtuality for simulating real world construction operations and scenarios in a 3D virtual world using a combination of Computer Aided Data (CAD) models, Geographic Information System (GIS), geometric proximity monitoring and sensor-based position-orientation updates.
- A 3D visualization system, SeePlusPlus based on the Hybrid Virtuality principle that uses the modules for proximity monitoring, sensor stream acquisition and allocation for providing visual assistance and audio-visual warnings

The individual chapters of the dissertation are divided such that each focuses on a specific research issue and its associated challenges. Their individual contributions are summarized next.

Chapter 2 - This chapter provided an overview of the problem and the proposed methodology of using real-time 3D visualization for concurrently simulating real-world operations. In this chapter the concept of Hybrid Virtual (HV) was introduced as a means to simulate/emulate real-world operations in a 3D virtual world using tracking and geographic information from the real world. The various components of an HV simulation framework were introduced and described in detail. The concept of variable degrees of real-world representation dependent on simulation goals was also introduced.

The viability of HV simulations to represent real-world operations was presented through initial tests of the framework. These tests used a combination of sensor input and user input to control an excavator model in a virtual scene. Results of the tests demonstrated the potential use of the HV framework to assist excavation crews in preventing buried utility strikes. The tests also showed the limitations of the current framework due to the accuracy of input elements such the utility input data, and GPS receiver. In addition to the above, the three main causes of continued unintentional strikes to buried utilities were identified in this chapter as: (1) the deficiency in accurate position and semantic information of buried utility data, (2) the absence of persistent visual guidance, and (3) the lack of real-time spatial awareness of excavator operators in relation to the proximity

of the digging implement to the underlying utilities. The applicability of HV simulation to prevent such excavation-related accidents by alleviating each of the above three problem areas was outlined in this chapter.

Chapter 3 - This chapter focused on the buried utility aspect of the proposed monitoring framework. In the creation of a 3D virtual world representing an excavation operation, the buried utilities form one half of the relationship that needs to be monitored in real-time, with the other half being the equipment end-effector. This chapter described the details of the various stages in the life-cycle of underground utility geospatial data, and analyzed the inherent limitations that preclude the effective use of the data in downstream engineering applications such as excavation guidance. Five key requirements—Interactivity, Information Richness, 3-Dimensionality, Accuracy Characterization, and Extensibility—were identified as necessary for the consumption of geospatial utility data in location-sensitive engineering applications.

A visualization framework named IDEAL that meets the outlined requirements to guide the design and implementation of specific applications was developed and presented. The framework was implemented through a real-time, user-interactive, 3D visualization. In addition, the creation of the georeferenced 3D models of buried utilities from XML data sources was described. The 3D models created are color-coded to the American Public Works Association standards, and contain attribute information and uncertainty buffers. Displaying the uncertainty associated with visualized utility location data is stated as

being a key element in any visual guidance provided to excavator operators or field personnel to minimize accidental damage.

Chapter 4 - The primary focus of this chapter was the proximity-monitoring framework that uses 3D models present in the virtual world together with their real-world position and orientation to compute Euclidean distance, and any collisions between them. In this chapter, the types of spatial conflicts occurring between entities on construction jobsites were investigated and the limitations in current methodologies for dealing with scenarios that involve concealed or buried infrastructure were identified. A computing framework based on position-orientation sensors' input and 3D virtual models—that provides distance, collision, and tolerance queries was introduced and described. The computation framework is designed to be able to work with a real-time 3D visualization scheme that provides equipment operators with job-site views that are not possible through conventional on-site cameras. Results of proximity measurement performance and latency tests carried out at the National Institute of Standards and Technology's Intelligent Sensing and Automation Testbed were also presented in this chapter.

Chapter 5 - The thrust of this chapter was on the dynamic half of the tracked relationship from the real world. The challenges involved in creating a seamless and accurate link between the real and virtual worlds were examined. In this chapter the types of equipment monitoring that occurs on construction and mining jobsites as well as in other commercial settings were investigated. The need for detailed or micro-level equipment monitoring was presented and the areas where its use can help reduce accidents and

improve overall safety were described. This chapter also presented a methodology for equipment monitoring based on 3D visualization and real-time proximity monitoring using sensor-based input for updating 3D equipment components. The principles laid out in the proposed methodology and technical approach were demonstrated through an interface for mapping sensor data streams to specific equipment components. This interface was presented in context of the overall 3D visualization application for assisting excavator operators for preventing unintended strikes with underground utilities.

The chapter also described validation experiments that demonstrated the ability to simulate the real world motion of equipment in a 3D virtual world. The experiments simulated the motion of a backhoe loader's articulated arm through orientation sensors installed on its boom, stick/dipper and bucket. The results comparing accuracies in real and virtual worlds were presented. In addition, the effect of a warning mechanism based on a preset safety threshold was investigated on the performance of the operator.

6.4 Future Directions of Research

This dissertation has presented a method for assisting excavation operators with real-time 3D visualization and audio-visual warnings for impending collisions. The various chapters in this dissertation have focused on individual frameworks and applications comprising the overall real-time visualization approach. Validation experiments for the individual frameworks are also described in chapters 2 through 5. The imminent opportunity for additional experiments is to investigate the effect of real-time visualization on excavator operators in actual excavation scenarios and the additional

challenges thereof. The effect of real-time visualization on operator performance can be measured through a NASA task load validation matrix as shown in Table 6.1.

Human Factors			
Criteria	**Assessment Factors**	**Assessment Methodology**	**Assessment Metrics (Very Low = 0; Very High = 10)**
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 or rushed was the pace of the task?	NASA Task Load Index guidelines	Very Low to Very High
Performance	How difficult was it to maintain a high productivity and performance level?	NASA Task Load Index guidelines	Very Low to Very High
Effort	How much effort was used to perform the task safely?	NASA Task Load Index guidelines	Very Low to Very High
Frustration	What level of difficulties were experienced while performing the task?	NASA Task Load Index guidelines	Very Low to Very High

Table 6.1: NASA Task Load Index validation matrix

Operators can be required to evaluate the six parameters of the index in a two step approach – first while performing excavation operations without real-time visualization support, and second while performing the similar activity with the support of real-time visualization. The validation experiments presented for individual frameworks present the

errors and uncertainty contributed by the position and orientation sensors as well as the proximity computations. However, the aggregation of uncertainties along the kinematic chain of components will result in an overall position and orientation uncertainty for the equipment end-effector. This end-effector uncertainty will be a function of the individual uncertainties of sensors placed on the equipment's kinematic chain such as GPS and accelerometers.

The magnitude of this end-effector safety buffer can increase or decrease the actual distance collision based upon the aggregation of errors. For example, an end-effector uncertainty buffer of 1.5 inches can increase or decrease the distance to collision by this value and thus the end-effector may be closer to colliding with an object such as a buried utility than may be computed by ignoring errors in sensor reported position and orientation. Future experiments performing aggregation of errors from the GPS (position) sensor, and accelerometers (orientation) sensors along the kinematic chain can provide warnings to operators after inclusion of end-effector position-orientation uncertainty.

While the primary focus of the research has been excavation related operations, the outcome is designed in such a manner that any typical construction operation can be monitored in real-time by providing operators with information support. In addition to the core functionality, the following areas are viewed as logical future avenues of research.

6.4.1 Trace Simulation (construction accident reconstruction)

The ability to represent ongoing construction operations in a 3D virtual world can be extended to recreate construction-related accidents. Through the use of position and orientation data logs from stored sensor data streams, effect of changes, and mapping of appropriate sensor stream to dynamic entity, the 3D visualization and animations can be used to reconstruct and replay an accident related event that occurred in the past (Cor 1998). However, it is important to note that the 3D visualization of the accident event is truly reliable only if the computations account for the laws of physics governing the motion of the object. This eliminates the possibility of depicting motions in the visualization that are not feasible in the real world. Integration of a physics-based computation engine such as the Open Dynamics Engine (ODE) into the existing 3D visualization and proximity monitoring framework can effectively allow playback of accident-related events to better understand them.

6.4.2 Intelligent Teleoperation for Equipment on a Construction Jobsite

In traditional teleoperation the task of an operator is similar to what they would perform if present in the equipment's cabin but typically from a remote location due to unsafe or risky work environments. In the case of 'Intelligent Teleoperation' the details of the task are abstracted away from the operator such that a single operator would be able to control multiple pieces of equipment at any given time. The role of an operator would thus be elevated to that of an 'operations director'. This can be achieved by setting up relationship pairs between an excavator and equipment when the operations start, using an interface present in the 3D visualization.

In the case of ‘Intelligent Teleoperation’, a two-way data transmission loop would be adopted. In addition to the existing real-to-virtual leg of transmission, as presented in the research thus far, the virtual world would transmit position-orientation requirements to equipment components in the real world to cause their required translation and/or rotation. The proposed methodology could use a combination of GPS and computer vision techniques together with inverse kinematics and proximity monitoring.

6.4.3 Human-in-the-loop simulation

A simulation model is referred to as Human-in-the-Loop (HITL) simulation if the model requires human interaction or participation in order to proceed or progress (DoD 1998). This research is an example of HITL simulation due to the operator controlling the equipment. However, the generic audio-visual warnings provided to the operator need to be optimized to take into account human (operator) behavior in order to be most effective and assist the operator in the best possible way to avoid a potential accident. Thus, the simulation of the real world operation needs to account for human factors in the simulation environment. The benefits of incorporating human factors in a simulation model have been experienced in fields such as Manufacturing to study the effect of worker behavior on a simulation model (Baines et al 2005) and Healthcare to study the effect of patient traits on a simulation model that determines the effectiveness of the healthcare on patients (Brailsford et al 2006).

The current approach of using position and orientation sensor updates to provide warnings is deterministic in nature. It also assumes that every operator will excavate the same amount of material and penetrate the same depth per stroke. It has been noted that the operating style and specifics vary from operator to operator (Hall 2003). In order to be able to provide warnings sufficiently in advance, a probabilistic approach may be required that takes into account the unique working style of each operator as well as the equipment characteristics. Such an approach would utilize the starting distance (depth) between the digging implement and utility and the depth penetrated per pass. The depth penetrated by the digging implement per pass is a function of the equipment as well as the operator. The use of machines that tailor themselves to the operator characteristics and set the safety threshold at a higher level as and when needed could be investigated. In this way the algorithm would account for operator-specific characteristics, ensuring that the warning will be given well in advance of a potential utility line strike.

6.5 References

Akinci, B. and Fischer, M. (2000). “4-D Workplanner: A Prototype System for Automated Generation of Construction Spaces and Analysis of Time-Space Conflicts”, Computing in Civil and Building Engineering (2000). September 2000, 740-747.

Allread, B. and Teizer, J. (2010). “Blind Spot Measurements for Real-Time Pro-Active Safety in Construction”, Construction Research Congress 2010. May 2010, 132-141.

Baines, T.S., Asch, L., Hadfield, L., Mason, J.P., Fletcher, S., Kay, J.M. (2005), “Towards a theoretical framework for human performance modelling within manufacturing systems design”, Simulation Modelling Practice and Theory, 13: 486-504.

Brailsford, S.C.; Sykes, J.; Harper, P.R. (2006), “Incorporating Human Behavior in Healthcare Simulation Models”, Simulation Conference, 2006, WSC 06, Proceedings of the Winter , vol., no., pp.466-472, 3-6 Dec. 2006.

Cheng, T., and Teizer, J. (2011). “Crane Operator Visibility of Ground Operations”, 2011 Proceedings of the 28th ISARC, Seoul, Korea, Pages 699-705.

Cor, H. (1998). "Using Simulation to Quantify the Impacts of Changes In Construction Work", Master's Thesis <http://scholar.lib.vt.edu/theses/available/etd-72598-145525/> (10/14/2012).

DoD (1998), “Modeling and Simulation (M&S) Glossary", DoD 5000.59-M, DoD, January 1998 <http://www.dtic.mil/whs/directives/corres/pdf/500059m.pdf> (10/14/2012).

Hall, A. (2003), “Characterizing the Operation of a Large Hydraulic Excavator”, Master’s Thesis, The University of Queensland, Brisbane, 2003.

Miss Dig Systems Inc. (2007), “Dig Safely – One Call Excavation Handbook” <http://missdig.net/images/2007onecall_handbook.pdf> (10/14/2012)

Son, H., Kim, Ch., Kim, H., Choi, K.-N., 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 (2008), pp. 232–237.

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

Appendices

Appendix A

Building Blocks for the Creation of a Hybrid Virtual Scene

This appendix describes all of the building blocks and other details required to create a Hybrid Virtual Scene (HVS). An HVS is a virtual representation of a real-world operation using 3D models to represent real-world entities, and sensor data from sensors recording the updated positions and orientations of the dynamic entities. This appendix provides details on the building blocks needed to create an HVS to represent an excavation operation.

The number of elements required to adequately represent an operation is determined by the intents of the scene modeler, and is guided by the requirements in the 'Level of World Representation' (LWR) concept. The LWR concept introduces a set of eight parameters that encompass all physical entities that are likely to be present on a construction jobsite. The types of entities range from static and dynamic to surface, sub-surface, and airborne objects. In the case of an excavation operation, the following six entities would typically be present on any jobsite and its immediate surroundings: 1) the excavator, 2) the buried utilities, 3) the jobsite terrain, 4) surrounding structures and building, 5) laborers and other workers, and 6) other by-standers. However, not all of the entities on a jobsite need to be modeled in the corresponding virtual world for the HVS to be effective for monitoring the key aspect of the operation.

The key interaction in the case of an excavation operation (i.e., the interaction that is being monitored to prevent accidents) is the one between the excavator's end-effector (the bucket) and the buried utilities. Thus, accordingly, the elements from the real world that are captured in the virtual world include the buried utilities, the excavator, and the jobsite terrain. Through the combination of using the LWR scale and analyzing the requirements of the HVS output, those elements from the real world that need to be represented in the virtual world can be determined by the scene modeler. The rest of this appendix provides details on the creation of HVS building blocks for the chosen elements (buried utilities, excavator, and jobsite terrain).

A.1 Jobsite Terrain

A 3D model representing the jobsite terrain is an essential requirement for the real-world operation in the virtual world. 3D Virtual environments are interactive in nature and allow the user to manipulate the view of the scene by changing the camera's position and/or orientation, and can lead to users losing their correct perspective of the scene. The terrain surface provides a reference point in the virtual scene and is thus viewed as an essential visual component. The 3D terrain model for an HVS is created using a combination of elevation data and georeferenced imagery representing the jobsite terrain surface, as shown in Figure A.1.

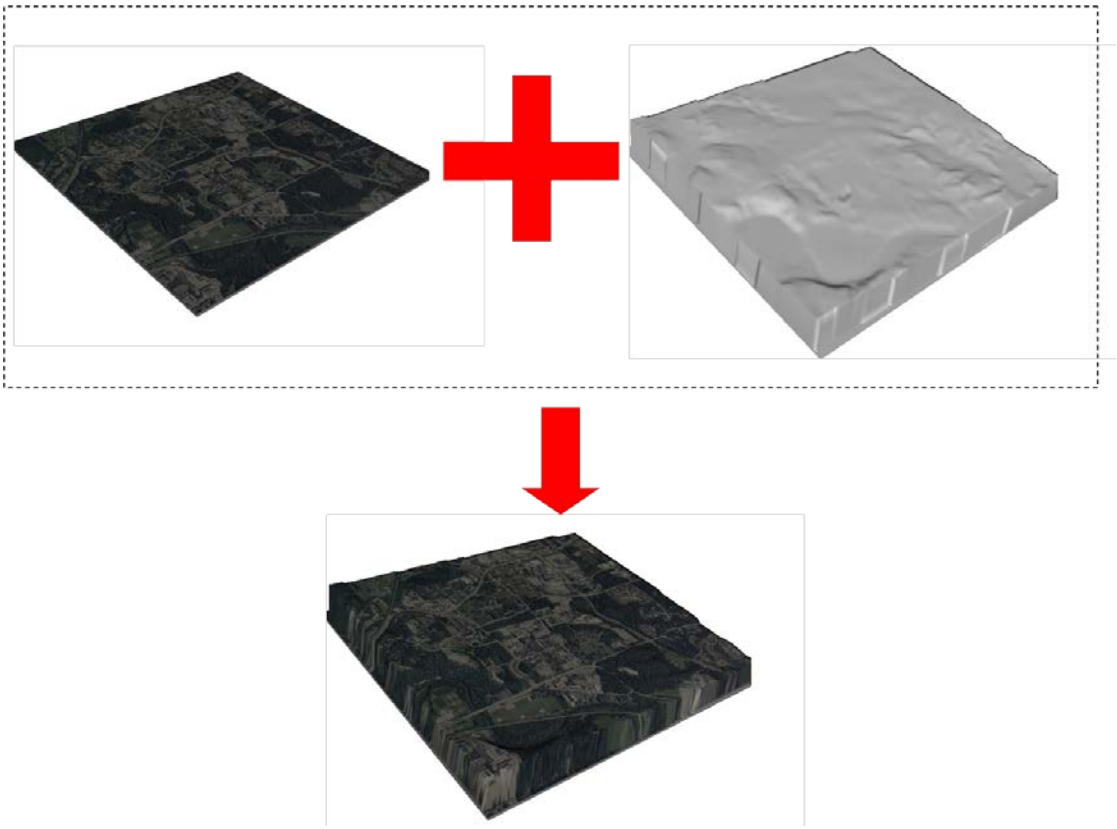

Fig. A.1: Creation of 3D terrain polygon model using elevation data and georeferenced imagery

A.1.1 Terrain Elevation Data

Elevation data for a terrain can be stored and represented in many formats, such as contour lines, triangular irregular networks (TINs), and raster formats. The raster form, which corresponds to a digital representation of the elevation of the terrain, lends itself effectively to create 3D terrain models. The most common raster format for representing terrain elevation is a Digital Elevation Model (DEM). A DEM refers to a regular grid (raster) of spot heights. DEMs are basically a sampled array of elevations for the ground position at regularly spaced intervals. The resolution value of a DEM is determined by the interval spacing between successive spot heights. Thus using a smaller-spaced,

higher-resolution DEM produces a more accurate 3D terrain model. DEMs are available from a number of sources, such as the United States Geological Survey (USGS) and other commercial repositories. The USGS has one of the largest repositories of elevation data in the entire United States, which is referred to as the National Elevation Dataset (NED). The accuracy of the data in the NED ranges from 3 to 30 meters for the conterminous United States.

The NED data details can be obtained from the USGS web repository located at http://ned.usgs.gov/. This repository contains elevation data for the entire United States. In order to locate the area of interest representing the jobsite, the "National Map Viewer and Download Platform" is used (http://viewer.nationalmap.gov/viewer/). This platform is an interactive tool used to download elevation, imagery, and other terrain-related data through a single cohesive interface, as shown in Figure A.2. The area of interest can be zoomed into using a GUI similar to other web-based map applications, and using the "Search" field to enter the jobsite address. An area surrounding the University of Michigan's (UofM's) Civil Engineering Department's GG Brown Building—located on UofM's North Campus region—is used in this example to demonstrate the process of obtaining elevation and imagery data. Figure A.3 shows a screen shot with the area of interest displayed by the viewer and the "Download Data" option highlighted.

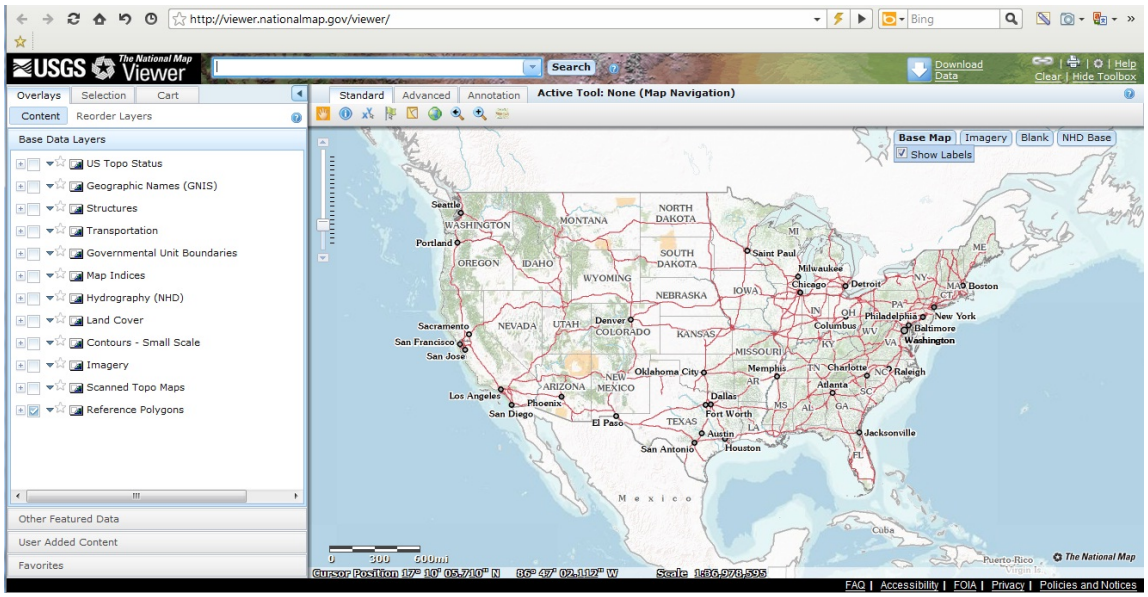

Fig. A.2: Screenshot showing National Map Viewer and Download Platform web interface

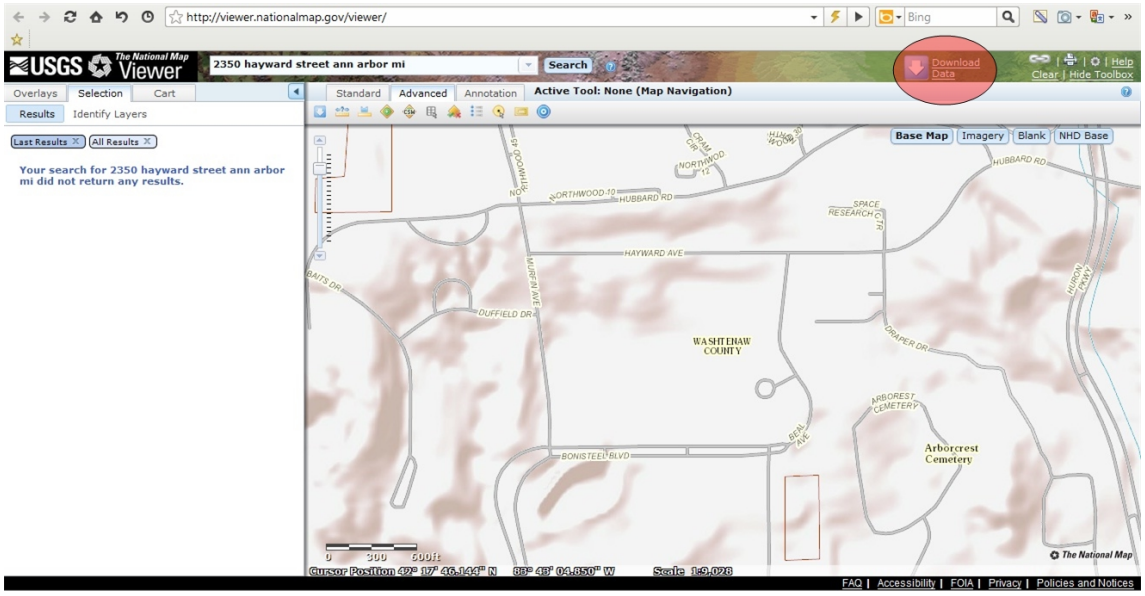

Fig. A.3: Screenshot showing jobsite/area of interest in focus of the National Map interface

The “Download Data” option in the interface presents the user with reference area options to download data. This is shown in Figure A.4 (1). By selecting “current map extent,” data will be downloaded for the area that is displayed by the National Map Viewer. After selecting the screen extent area, the user is then presented with a dialog box with a list of available data from USGS. This screen is shown in Figure A.4 (2) with the options for “Elevation” and “Orthoimagery” checked for download.

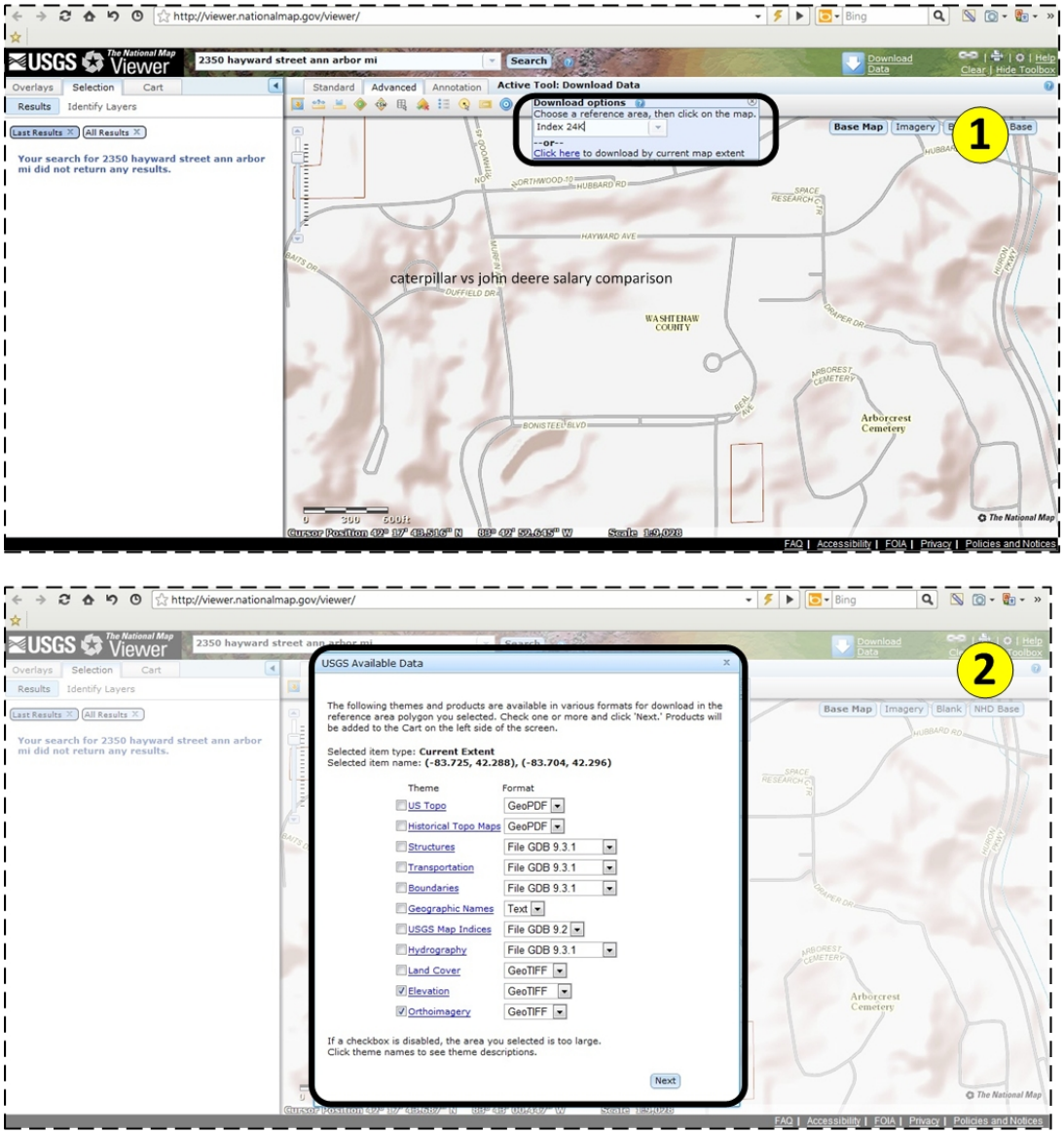

Fig. A.4: Steps 1 and 2 for downloading elevation and imagery data

The download options presented in Figure A.4 (2) also display the available file formats for a given data option, such as elevation. The file format “GeoTIFF” is selected for downloading Elevation (and Orthoimagery) data. The GeoTIFF format allows georeferencing information to be embedded within a regular TIFF file. It follows the requirements of the TIFF 6.0 format, hence it can be viewed and manipulated using the same software that regular TIFF images use. After the selection of the required data from the dialog in Figure A.4 (2), users will see displayed all of the available data products, having varying levels of quality, corresponding to the selected area and data type. The product options presented for the selected elevation data are shown in Figure A.5. The products are differentiated by their month and year of collection, their resolution, the data type they represent, and their corresponding metadata.

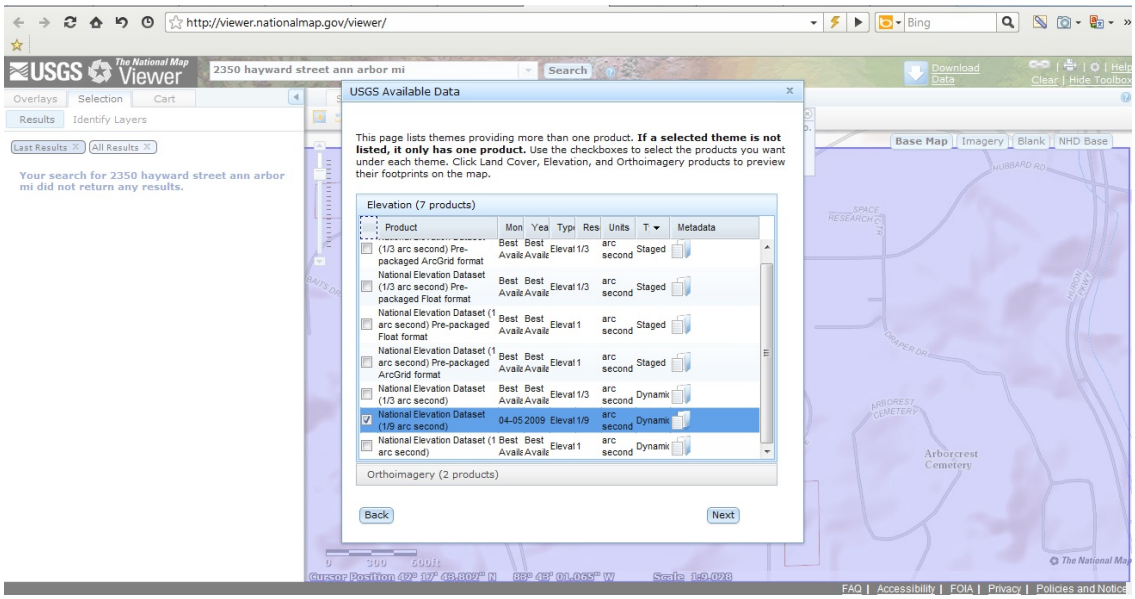

Fig. A.5: Available products options for elevation data corresponding to selected area

The best available elevation product can be determined by selecting the most recent product, and the one having the highest resolution. After the products are selected in the dialog from Figure A.5, the selected products are displayed in the National Map interface with an option to “Checkout,” as highlighted in Figure A.6. Also, Figure A.6 shows the selected products for both elevation and orthoimagery that are selected by the user. After the user confirms the ‘transaction’, they are requested to enter an email address to which the data download links are sent, and the user is subsequently presented with a confirmation dialog box, as shown in Figure A.7. The step corresponding to Figure A.7 represents the final stage of using the National Map interface. All of the following steps are carried out on the data that is downloaded directly to the user’s machine.

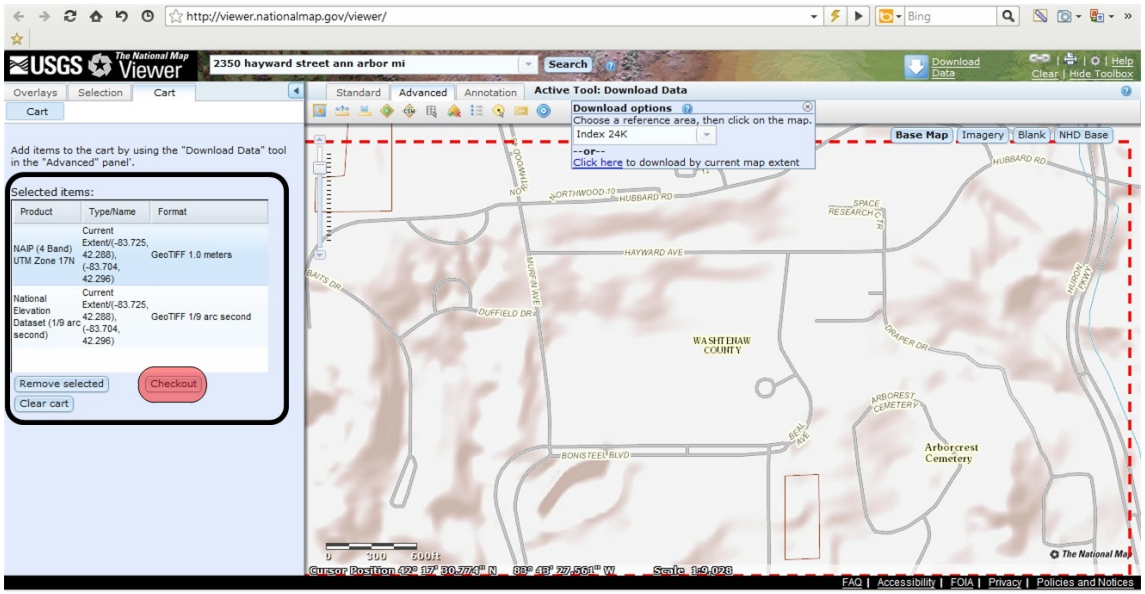

Fig. A.6: Data products checkout screen in the National Map interface

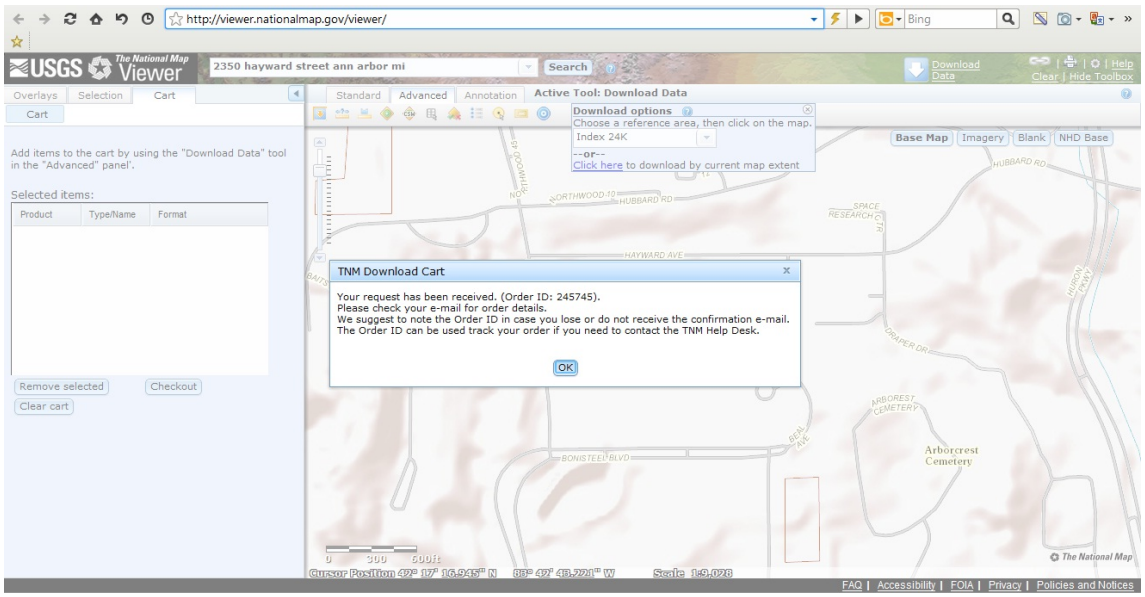

Fig. A.7: Confirmation of data products sent to the user-entered email address

A.1.2 Terrain Orthoimagery Data

The second requirement for a 3D terrain model is a georeferenced image that is an aerial image of the terrain. A georeferenced image is one that is location aware, that is, its extents—North and South, and East and West—are computed in relation to a geographic or projected coordinate system. The use of georeferenced imagery results in the ability to use the subsequent terrain model together with other entities from the real world that have their locations computed in a similar coordinate system as the orthoimagery and terrain.

Figure A.4 (2) presents the option to users to select orthoimagery data along with multiple file type options. As with elevation, GeoTIFF is selected as the file type for orthoimagery. This selection generates a list of data products for orthoimagery as shown in Figure A.8. Similar to the elevation-related data products, orthoimagery products are primarily differentiated by their date of collection, precision in terms of resolution, and

their metadata. After the selection of the orthoimagery product, the user is presented with a confirmation screen, as shown in Figure A.6, which allows users to confirm the data request. Confirming the request results in the data download links being sent to the user-specified email address. As the interface allows multiple data types to be selected simultaneously, it is recommended that the elevation and orthoimagery data be selected for a given terrain location at the same time, to ensure the correct overlap of the two layers.

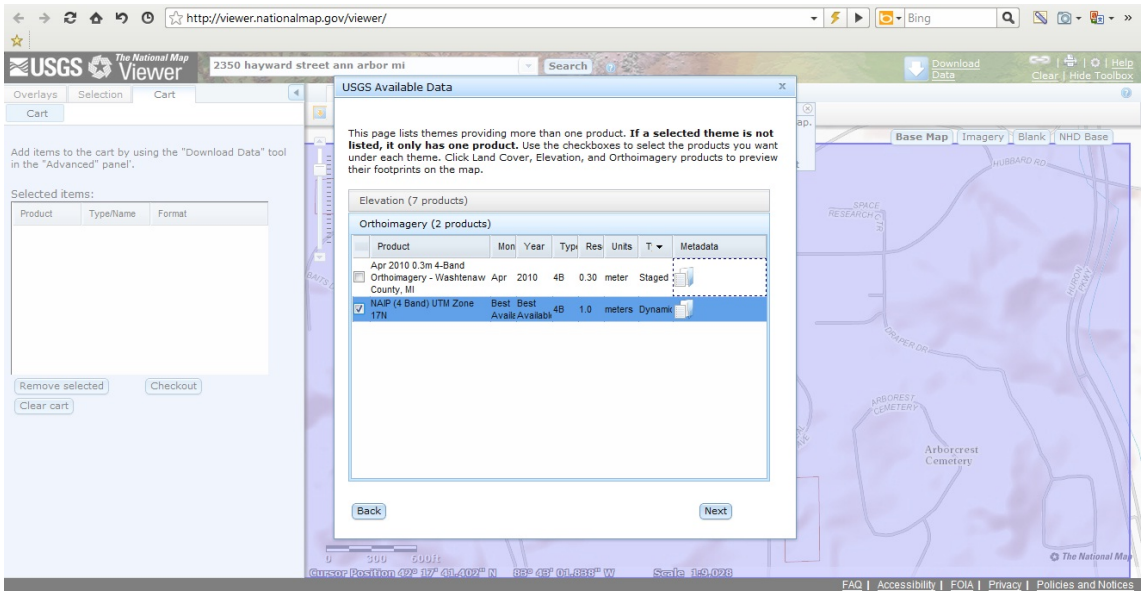

Fig. A.8: Available product options for orthoimagery data corresponding to selected area

A.1.3 Analyzing and Processing the Elevation and Terrain Data

The National Map interface sends the data download links for the selected products to the user-entered email address. The first step prior to downloading data from the USGS server is the creation of a “Terrain” folder with an identifier for distinguishing the jobsite it represents from other Terrain folders that will be created in future. The selected data

products are downloaded from the USGS server and extracted to the Terrain folder that was created. The current USGS setup means that downloaded elevation data and terrain data do not have clear identifiers in their folder and file names to distinguish between the two. However, the folder containing elevation data will have a file ("NED DataDictionary.pdf") that is not present in the orthoimagery data folder. Thus the two data folders can be distinguished from each other.

It is important to point out that the elevation and orthoimagery data downloaded from the USGS server can have different coordinate systems. The elevation data and image must have the same datum, projection coordinate system, and units for them to coincide. In order to make the coordinate systems of the elevation and orthoimagery layers, the coordinate systems need to be transformed to a common system. This process is done through the FWTools toolkit. FWTools is a set of Open Source GIS binaries for Windows and Linux (http://fwtools.maptools.org/). The "gdalinfo" tool within the suite provides information about the raster dataset, such as the raster extents, the coordinate system, and other information.

The output from gdalinfo on the orthoimagery raster is shown in the code snippet below. The output represents a selection of the complete output from gdalinfo but displays the coordinate system of the raster. In the case of the orthoimagery raster, the coordinate system is UTM 17N, NAD 83.

```
Driver: GTiff/GeoTIFF
Files: E:\Terrain_VPB_Demo\20943487\20943487.tif
       E:\Terrain_VPB_Demo\20943487\20943487.tfw
Size is 1760, 944
Coordinate System is:
PROJCS["NAD83 / UTM zone 17N",
    GEOGCS["NAD83",
        DATUM["North_American_Datum_1983",
            SPHEROID["GRS 1980",6378137,298.2572221010002,
                AUTHORITY["EPSG","7019"]],
            AUTHORITY["EPSG","6269"]],
        PRIMEM["Greenwich",0],
        UNIT["degree",0.0174532925199433],
        AUTHORITY["EPSG","4269"]],
    PROJECTION["Transverse_Mercator"],
    PARAMETER["latitude_of_origin",0],
    PARAMETER["central_meridian",-81],
    PARAMETER["scale_factor",0.9996],
    PARAMETER["false_easting",500000],
    PARAMETER["false_northing",0],
    UNIT["metre",1,
        AUTHORITY["EPSG","9001"]],
    AUTHORITY["EPSG","26917"]]
Origin = (275334.308000000020000,4686237.929999999700000)
Pixel Size = (0.999587431000506,-0.999587431000506)
Corner Coordinates:
Upper Left  (  275334.308, 4686237.930) ( 83d43'31.24"W, 42d17'45.57"N)
Lower Left  (  275334.308, 4685294.319) ( 83d43'29.92"W, 42d17'15.01"N)
Upper Right (  277093.582, 4686237.930) ( 83d42'14.50"W, 42d17'47.39"N)
Lower Right (  277093.582, 4685294.319) ( 83d42'13.19"W, 42d17'16.83"N)
Center      (  276213.945, 4685766.125) ( 83d42'52.21"W, 42d17'31.20"N)
```

On the other hand, the gdalinfo output for the elevation raster layer shows the following:

```
Driver: GTiff/GeoTIFF
Files: E:\Terrain_VPB_Demo\14472605\14472605.tif
       E:\Terrain_VPB_Demo\14472605\14472605.tfw
Size is 681, 260
Coordinate System is:
GEOGCS["NAD83",
    DATUM["North_American_Datum_1983",
        SPHEROID["GRS 1980",6378137,298.2572221010002,
            AUTHORITY["EPSG","7019"]],
        AUTHORITY["EPSG","6269"]],
    PRIMEM["Greenwich",0],
    UNIT["degree",0.0174532925199433],
    AUTHORITY["EPSG","4269"]]
Origin = (-83.725003181307869,42.296001719165396)
Pixel Size = (0.000030864197484,-0.000030864197484)
```

```
Corner Coordinates:
Upper Left  ( -83.7250032,  42.2960017) ( 83d43'30.01"W, 42d17'45.61"N)
Lower Left  ( -83.7250032,  42.2879770) ( 83d43'30.01"W, 42d17'16.72"N)
Upper Right ( -83.7039847,  42.2960017) ( 83d42'14.34"W, 42d17'45.61"N)
Lower Right ( -83.7039847,  42.2879770) ( 83d42'14.34"W, 42d17'16.72"N)
Center      ( -83.7144939,  42.2919894) ( 83d42'52.18"W, 42d17'31.16"N)
```

It can be observed that the elevation layer has a Geographic Coordinate System (GCS), while the orthoimagery layer has a Project Coordinate System (PCS). As the jobsite terrain represents a small area where distances between objects are measured in linear terms (such as in a PCS), re-projection of the elevation layer from GCS to PCS enables the entire terrain model to be in PCS, ensuring the overlap of the two data layers. The gdaltransform tool within the FWTools suite is used to re-project a raster data set from one coordinate system to another (http://www.gdal.org/gdaltransform.html). In the case of the data presented in this appendix, the elevation layer is reprojected from GCS (NAD83) to PCS (UTM 17N, NAD83). Once re-projected, the two layers can be used in the following step of creating 3D terrain models.

A.1.1 Virtual Planet Builder / osgEarth

Virtual Planet Builder (http://www.openscenegraph.org/projects/VirtualPlanetBuilder) (VPB) and osgEarth (http://www.osgearth.org/) are two commonly used 3D terrain creation toolkits. Both of these toolkits have been successfully used to create 3D terrains for SeePlusPlus. VPB creates a 3D terrain model file that is stored in memory. This file is a sub-graph that is added to the HV scene when created. OsgEarth, on the other hand, uses an XML file to reference the elevation and image data. The terrain is rendered at runtime and, unlike VPB, the 3D terrain file is not physically present on the hard disk.

Both toolkits are based upon the OSG framework and thus seamlessly integrate with any OSG-based application, such as SeePlusPlus. The elevation data is processed by the terrain-creating algorithm that creates a 3D mesh representing the real-world terrain using the spot heights from the DEM. The polygon mesh is then draped with the georeferenced image of the terrain to give a complete 3D terrain. The 3D terrain model produced by the VPB application from OSG is shown in Figure A.9.

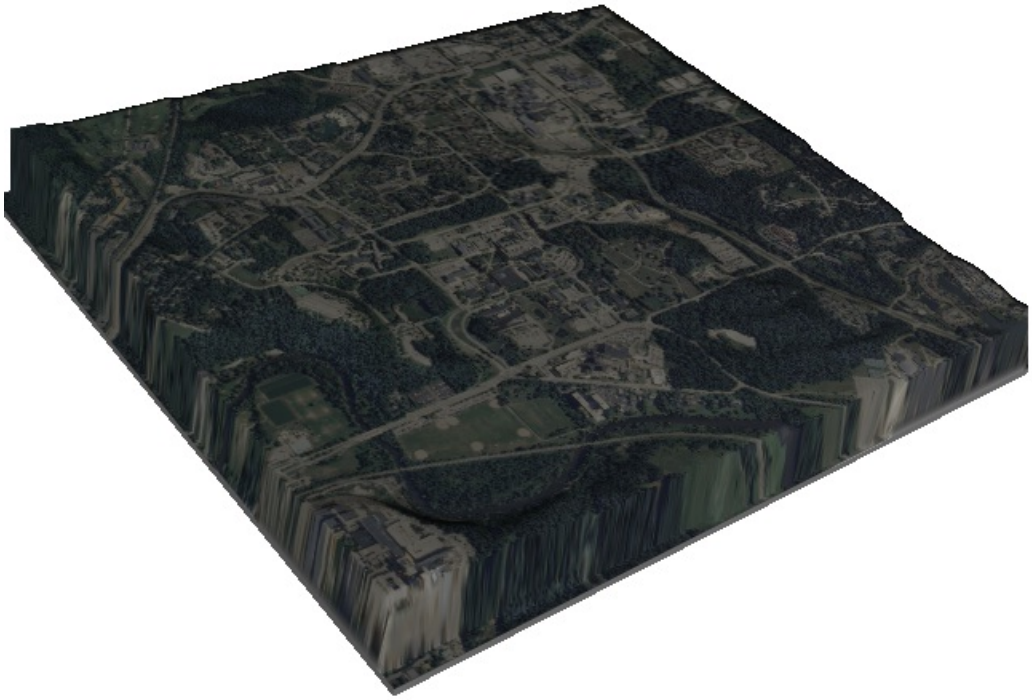

Fig. A.9: 3D terrain model produced by Virtual Planet Builder application (from Open Scene Graph)

A.2 Utilities

The next building block for creating an HVS representing excavation operations is the underground infrastructure or buried utilities. Buried utility data is stored in multiple formats such as Geographic Information Systems (GIS), CAD, and XML. Of all the data formats, XML presents the highest level of flexibility to the user for areas such as the inclusion of specific attribute information, accuracy and quality information of buried utilities. This is primarily due to its flexibility for the creation of user-defined data tags that can encode any information within the XML document, which users determine as being essential pieces of information. In this section, the details on creating 3D models for buried utilities from GPS data and subsequent XML files are provided.

The data flow for buried utility information is presented in Figure A.10. GPS location data is combined with user-selected attribute information in the B3M Collector tool to produce XML files that contain all of the location, accuracy, and attribute information. These XML files are then processed by the B3M Creator tool to produce georeferenced 3D models. The 3D utility models are provided with the same coordinate system as the terrain model to ensure that the different building blocks can be used in the same HVS. The 3D utility models are also color coded according to the utility type, satisfying the requirements of the American Public Works Association (APWA), and have attribute information embedded in the model that is displayed to the user during the visualization stage of the HVS.

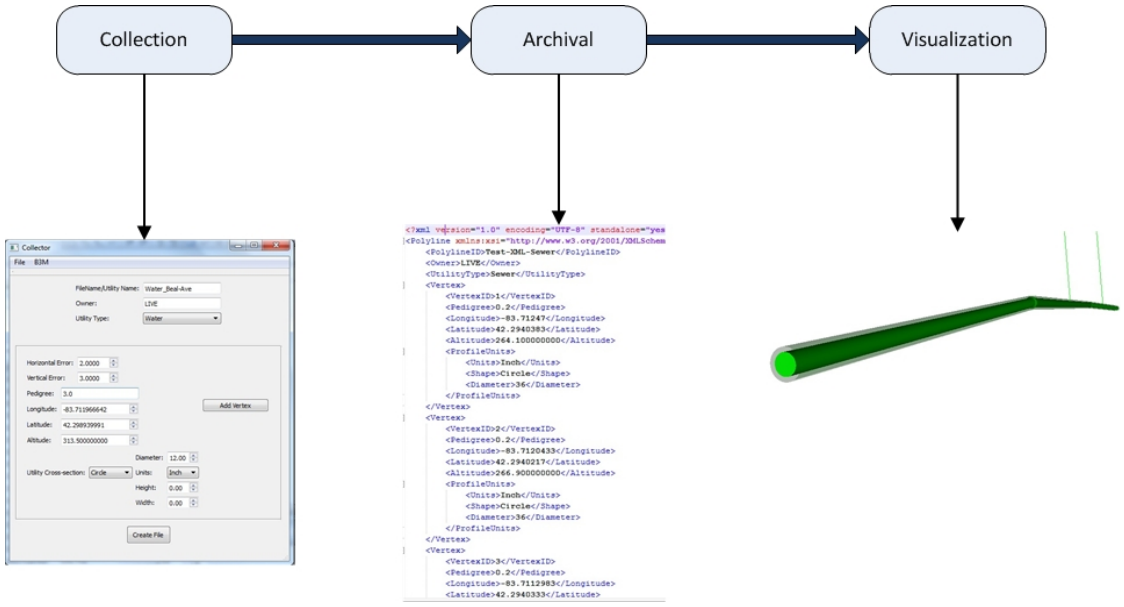

Fig. A.10 Buried utility data flow through collection, archival, and 3D modeling stages

A.2.1 B3M Collector

B3M Collector's graphical user interface (GUI) is shown in Figure A.11. The basic purpose of this tool, as its name suggests, is to collect data from the field and archive it as XML files such that the data gets post-processed for the creation of 3D models. If a user has pre-existing XML files representing buried utilities, the collector step can be skipped and the 3D models can be directly generated for these files. B3M Collector is designed to be used in combination with a GPS receiver. Some older GPS units may require the use of a serial-to-usb port adapter. After a GPS receiver is connected to the user's data collection PC, the B3M Collector application can be started. Once the GPS receiver is detected, the GPS-dependent values, as shown in Figure A.11, get updated in real time. The application is designed to allow users to generate a unique XML file corresponding to a utility line in the field.

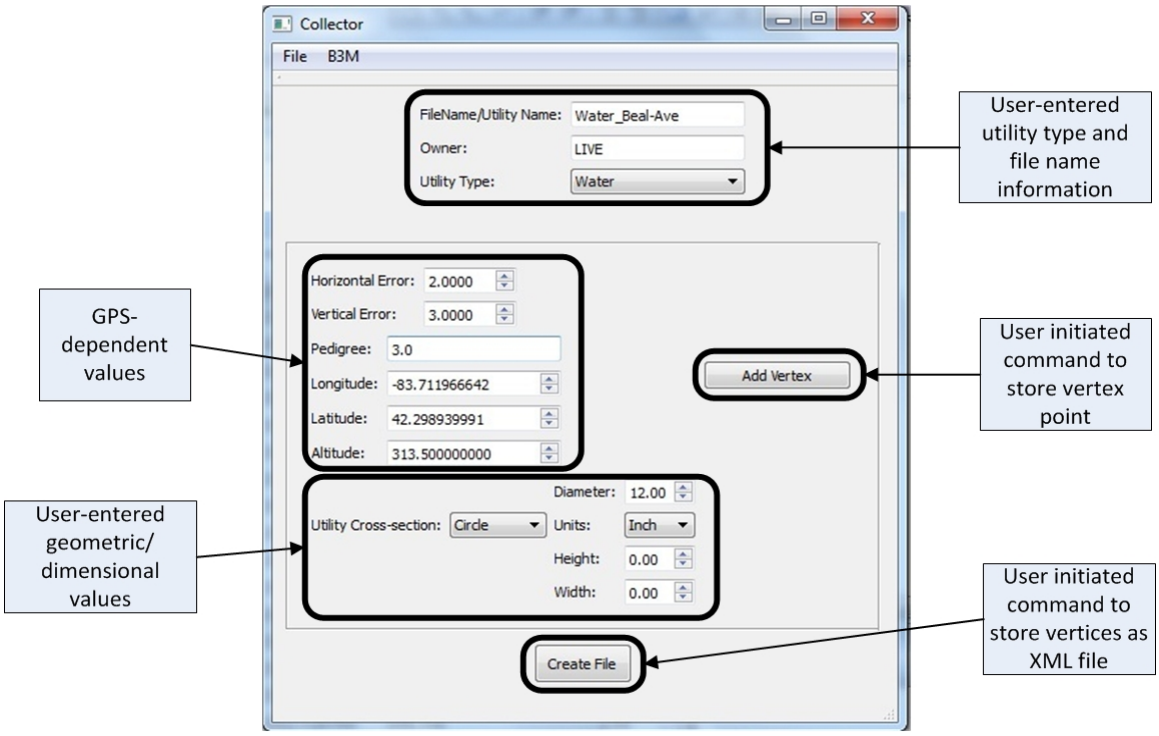

Fig. A.11: User interface for collecting buried utility location, attribute, and accuracy data using user input and GPS receiver information

Data is collected for every vertex or bend of a given utility line where the direction changes in either the horizontal or vertical plane. As shown in Figure A.12, there are four vertices where the water utility line has a change in direction. The user would position the GPS such that the elevation of the top surface of the utility would be recorded when the user clicks on the "Add Vertex" command on the B3M Collector interface. Every Add Vertex command triggered by the user results in the geographic location (latitude, longitude, altitude, position accuracy) of that point recorded. In addition, the physical details of the utility line—such as its geometric dimensions, utility type, and its owner—are also recorded in the XML file that is generated and archived.

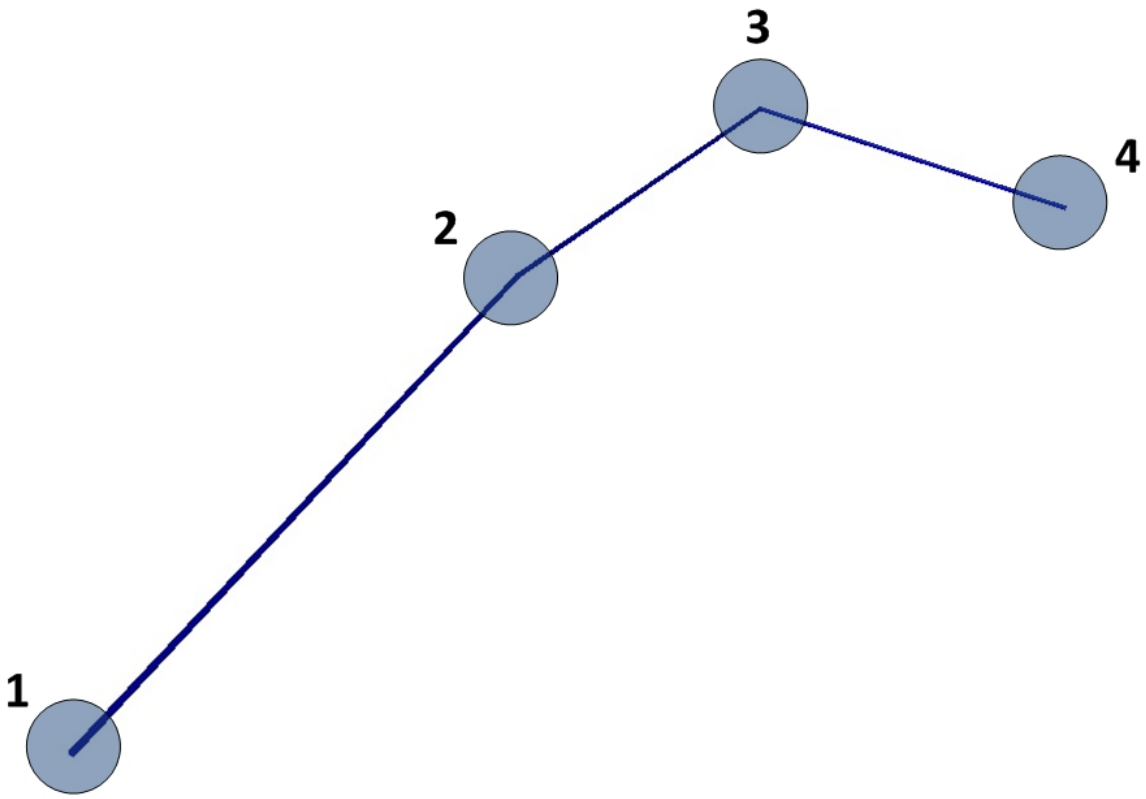

Fig. A.12: Water utility line with four vertices where data collection is required

The XML file generated for the water utility line shown in Figure A.12 is shown in the following code sample. The primary attributes stored in the XML file are the Utility ID (polyline ID), the utility owner, and the utility type. In addition, every vertex has location and geometric attributes associated with it. The interface allows users to specify the cross-sectional shape of the utility line, with the two options being circular and rectangular. In addition, the units used for storing the geometric dimensions are also stored in the XML file.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Polyline          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="LIVE_Schema.xsd">
    <PolylineID>Test-XML-water</PolylineID>
    <Owner>LIVE</Owner>
    <UtilityType>Water</UtilityType>
```

```
    <Vertex>
        <VertexID>1</VertexID>
        <Pedigree>1.0</Pedigree>
        <Longitude>-83.71192</Longitude>
        <Latitude>42.2949167</Latitude>
        <Altitude>269.000000000</Altitude>
        <ProfileUnits>
            <Units>Inch</Units>
            <Shape>Circle</Shape>
            <Diameter>24</Diameter>
        </ProfileUnits>
    </Vertex>
    <Vertex>
        <VertexID>2</VertexID>
        <Pedigree>1.0</Pedigree>
        <Longitude>-83.71096</Longitude>
        <Latitude>42.2949533</Latitude>
        <Altitude>271.300000000</Altitude>
        <ProfileUnits>
            <Units>Inch</Units>
            <Shape>Circle</Shape>
            <Diameter>24</Diameter>
        </ProfileUnits>
    </Vertex>
    <Vertex>
        <VertexID>3</VertexID>
        <Pedigree>1.0</Pedigree>
        <Longitude>-83.710145</Longitude>
        <Latitude>42.294895</Latitude>
        <Altitude>270.700000000</Altitude>
        <ProfileUnits>
            <Units>Inch</Units>
            <Shape>Circle</Shape>
            <Diameter>24</Diameter>
        </ProfileUnits>
    </Vertex>
    <Vertex>
        <VertexID>4</VertexID>
        <Pedigree>1.0</Pedigree>
        <Longitude>-83.71011</Longitude>
        <Latitude>42.294475</Latitude>
        <Altitude>270.800000000</Altitude>
        <ProfileUnits>
            <Units>Inch</Units>
            <Shape>Circle</Shape>
            <Diameter>24</Diameter>
        </ProfileUnits>
    </Vertex>
</Polyline>
```

A unique XML file, such as the one seen above, is generated by the B3M Collector application for every new utility line that the user collects data for. Such XML documents

are referred to as XML instance documents. The user-defined tags in XML instance documents conform to a pre-defined schema document. An XML schema defines the elements and their order in an instance document, and is used to ensure that an instance document confirms to the preset structure. The data in instance documents is extracted through a process called ‘parsing’, which takes place in the 3D modeling stage. Instance documents are checked for conformance with the schema document to ensure that data received by the parsing stage is valid.

Instance documents have the file extension ‘.xml’ while schema documents have a ‘.xsd’ file extension to help users differentiate between the two. The schema document is predefined for use in the XML file generation by B3M Collector, as well as the parsing stage during the 3D modeling stage in the following B3M Creator application. The schema document includes all of the data tags shown in the XML instance document (i.e., those that are determined to be required for accurate 3D model generation and adequate attribute display during visualization).

A.2.2 B3M Creator

The next step is the creation of the 3D models of utilities from the XML files that were generated in B3M Collector. It is important to point out that B3M Collector is designed to automatically generate 3D models of utilities for every saved XML instance file by calling the B3M Creator application internally. B3M Creator can also be used independently of B3M Collector by supplying the necessary XML instance files. B3M

Creator is based on scene graph architecture and is implemented using Open Scene Graph (OSG).

B3M Creator creates georeferenced 3D models of buried utilities that are color coded according to the APWA standards using the utility type information entered by the user in the B3M Collector interface. In addition, an uncertainty buffer is created around the buried utility models. This uncertainty buffer corresponds to the value of the "Pedigree" element in the XML instance file. Pedigree is recorded on a per vertex basis and is based upon the precision of the GPS coordinates at a given vertex. The uncertainty buffer is modeled as a translucent cylinder around the actual 3D model corresponding to the utility. The utility attribute information that is recorded in B3M Collector is also embedded into the 3D utility models by the B3M Creator process. An example of a 3D model produced by B3M Creator for the utility line shown in Figure A.12 is presented in Figure A.13.

In the case where pre-existing XML data is present or XML instance files are obtained from a database conforming to the schema used in B3M Collector, B3M Creator can be used as a standalone application to generate georeferenced 3D models of buried utilities. The application is designed with a command line interface to enable the batch processing of multiple instance documents. An example of executing the B3M_Creator application with two XML instance documents is shown in the code sample below. The 3D models (for gas and electric utilities) produced from the execution are shown in Figure A.14.

```
B3M_Creator.exe Test-XML-Electricity.xml Test-XML-Gas.xml
```

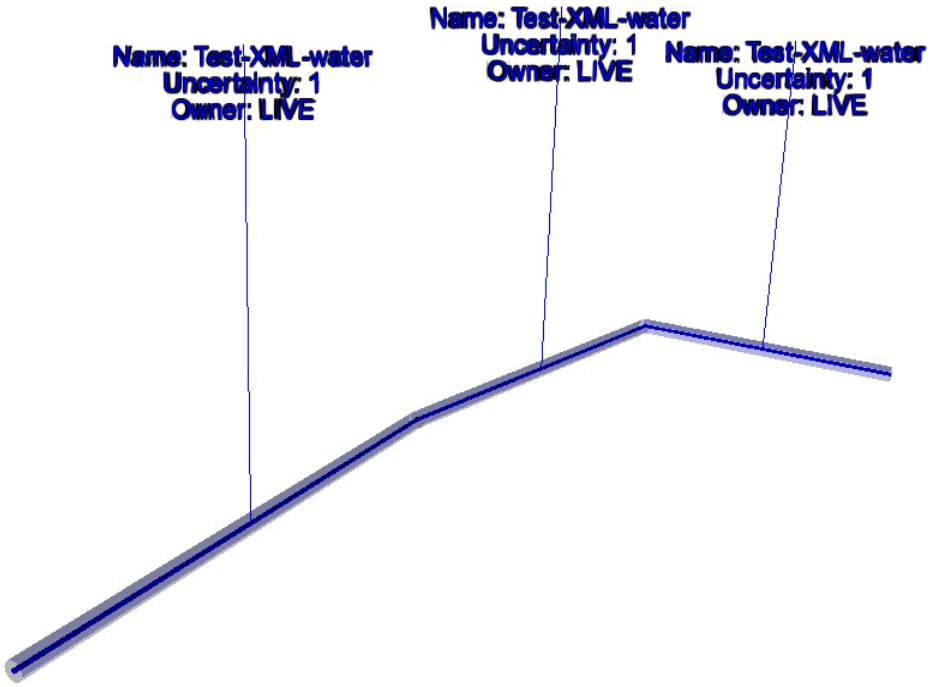

Fig. A.13: Buried utility 3D model for a water utility line produced in B3M Creator

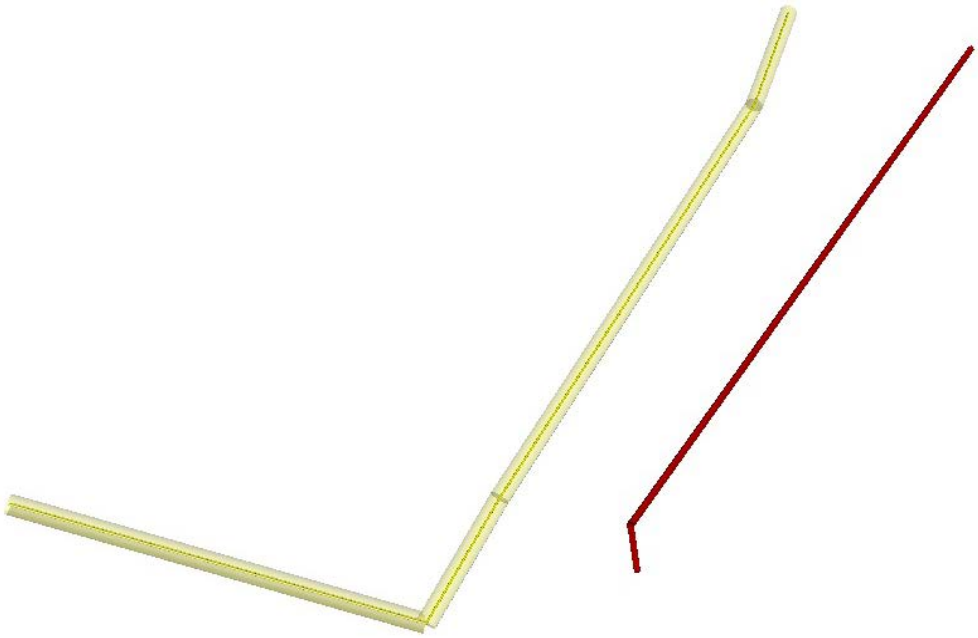

Fig. A.14: Gas and electric 3D utility models produced from the execution of the B3M_Creator

As already stated, B3M Creator is implemented using OSG and uses OSG's binary native file format (osgb) to create and archive 3D utility models. The filename for the created 3D model is derived from the user-supplied utility name during the B3M Collector stage or as obtained from the PolylineID data tag in the XML document. Thus the input files 'Test-XML-Electricity.xml' and 'Test-XML-Gas.xml' produce 3D model files 'Test-XML-Electricity.osgb' and 'Test-XML-Gas.osgb', respectively. These files can be used in any 3D visualization application that is based on OSG, as is described in Appendix B.

B3M Creator generates two 3D model files as output for every XML instance file provided as input. The first file type (I) represents the complete 3D model of the utility, including the uncertainty buffer, attributes, and the segments representing the actual utility. The second file type (II) corresponds to the 3D model that would be used for geometric proximity analysis, and includes only those geometric elements that are required for carrying out accurate collision detection and proximity computations. Thus for the input files 'Test-XML-Electricity.xml' and 'Test-XML-Gas.xml', the 3D model files produced for use in collision detection and proximity monitoring are 'Test-XML-ElectricityHalo.osgb' and 'Test-XML-GasHalo.osgb', respectively.

The type I of 3D model files produced by B3M Creator is intended to be used in 3D visualization applications, as they contain all visual elements that are required to convey the complete information to end users. File type II produced by B3M Creator is further processed to generate a 'tris' file that represents the list of geometric primitives making

up the 3D model. The use of tris files in proximity and collision query generation is described in Appendix B.3.

A.3 Equipment

The third and final building block for creating an HVS representing excavation operations is the equipment used on the jobsite. In the case of excavation operations, this equipment is an excavator. The georeferenced buried utility models represent the static elements in the HVS, while the excavator model in the HVS corresponds to the dynamic entity on the jobsite. The translation and rotation motions of the excavator in the real world, when replicated in the HVS, enable proximity and distance queries that provide spatial information along with real-time 3D visualization assistance to users.

Construction equipment like excavators, graders, backhoe, loaders, cranes, and dump trucks are made up of multiple smaller components that combine together in a parent-child hierarchical relationship. Using an excavator as an example, the components making up the overall equipment are the crawler, cabin, boom, stick, and bucket. The crawler is the parent component of all the other components, and the rotation or translation of the parent component results in the equivalent motion for the child components. In the case of the boom, stick, and bucket, the individual components are linked through joints that allow rotation about their pivot. In order for the HVS to be an accurate representation of the real world, the dimensions and behavior of the real-world and virtual-world equipment must correlate to each other.

Dimensional equivalency of the equipment components in the real and virtual worlds is a primary requirement of the 3D-equipment-modeling step. This feature is provided to a user through the 'Dimensioner' application that allows any equipment component to be re-dimensioned such that the modified component has the exact dimensions of its real-world counterpart. The second requirement is the ability to capture the motion of the equipment in the real world in the virtual equipment model. The parent-child relationship where the overall 3D equipment model is built up from existing individual 3D components is implemented through the Virtual Equipment Builder. The following sections provide details on these applications.

A.3.1 CAD models dimensioning for equivalence

The dimensioning of CAD models (equipment components) is made possible through the 'Dimensioner' application. This application provides a command line interface to users where an existing equipment component is provided as input for the application. The application computes the extents (dimensions) of the input component in the X-, Y-, and Z-directions, which correspond to length, width, and height dimensions. Table A.1 provides an example of a scenario where the existing and required dimensions of the equipment component differed from one another. Thus the Dimensioner application would be used to re-size the component, with the user inputting the required dimensions of the component. It is important to point out that the Dimensioner application computes all extent values in meters.

	X-extent (m)	Y-extent (m)	Z-extent (m)
Original boom component dimensions	12.77	1.4	3.72
Required boom component dimensions	2.75	0.37	0.79

Table A.1: Comparison of existing and required boom component's dimensions in X-, Y-, and Z-extents (i.e., length, width, and height)

Thus it is clear from Table A.1 that the existing boom component needs to be downsized before it can be used as part of the equipment to be created in the Virtual Equipment Builder application. The existing boom component is provided as input to the Dimensioner application that has a command line interface. The command line output is generated as follows:

```
>Dimensioner.exe "E:\Projects\SeePlusPlus\Backhoe\_BHBoom_.lwo"
X-Extent = 12.7725      Y-Extent = 1.4  Z-Extent = 3.71631
Enter the required extents along X, Y, Z axes each  separated by a
space:
2.75 0.37 0.79
Do you want the scaled model to be translucent? Y|N
N
```

The application displays the length, width, and height of the input component and resizes the component to the dimensions (in meters) provided by the user. The application also requests user input for specifying the transparency of the resized component. A translucent equipment component is used in those cases where a positional uncertainty exists in relation to the component. This feature has been implemented to demonstrate an uncertainty buffer around the bucket of an excavator where the translucent buffer around the bucket represents position and orientation uncertainties associated with computation of the bucket's global position and orientation.

The Dimensioner application also provides a preview of the resized component to the user, as shown in Figure A.15. The application also renames the modified component model and creates an archived copy. For example, the boom file '_BHBoom_.lwo' provided as input to Dimensioner, will be processed and saved as '_BHBoom_.lwoScaled.osgb'. Thus the resized components are saved in OSG's native binary format and can be with the terms 'Scaled.osgb' appended to the end of the input filename.

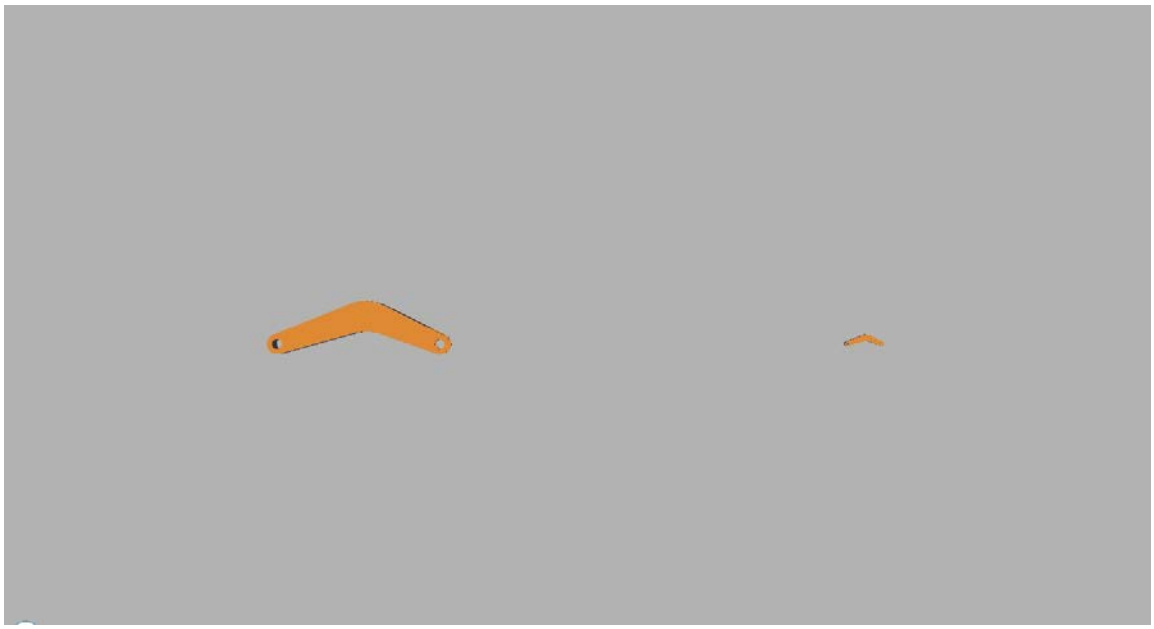

Fig. A.15: Resized equipment components preview after processing in the Dimensioner application

A.3.2] Virtual Equipment Builder (VEB)

The VEB application allows users to combine the individual equipment components into a single equipment model that can then be used in real-time 3D visualization applications such as SeePlusPlus, as explained in Appendix B. VEB is designed and implemented as an interactive graphical user interface, as shown in Figure A.16. The commands for building and archiving 3D equipment models are described in this section. The VEB application usage is demonstrated through an example of a 3D excavator that is created from its individual (disparate) components, as seen in Figure A.17.

Fig. A.16: Virtual Equipment Builder user interface showing front, left, top, and perspective views

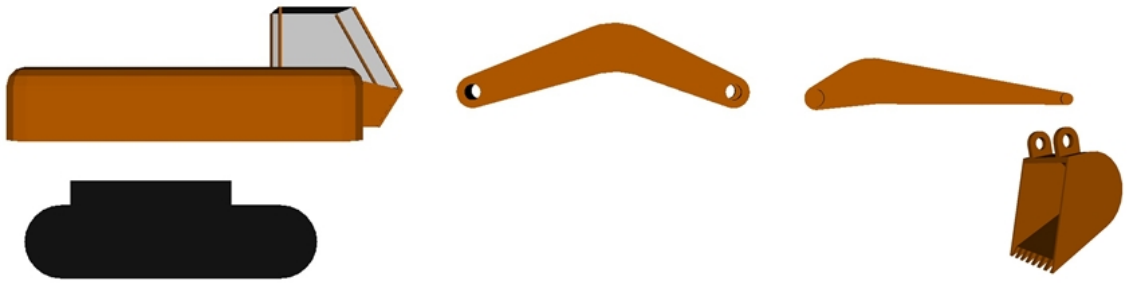

Fig. A.17: Individual disparate components (track, cabin, boom, stick, and bucket) used to build virtual excavator equipment

Before VEB can be used, all equipment components need to be resized to match the dimensions of their real-world counterparts using the Dimensioner application. Once this step is complete, the building process is initiated through the ‘Build’ command under the ‘Object’ menu, as shown in Figure A.18.

Fig. A.18: Build command highlighted under the Object menu of VEB

Execution of the Build command presents an input dialog box to the user in which the name of the equipment object being built is entered. This is shown in Figure A.19 with the ‘Excavator_1’ object being built in VEB.

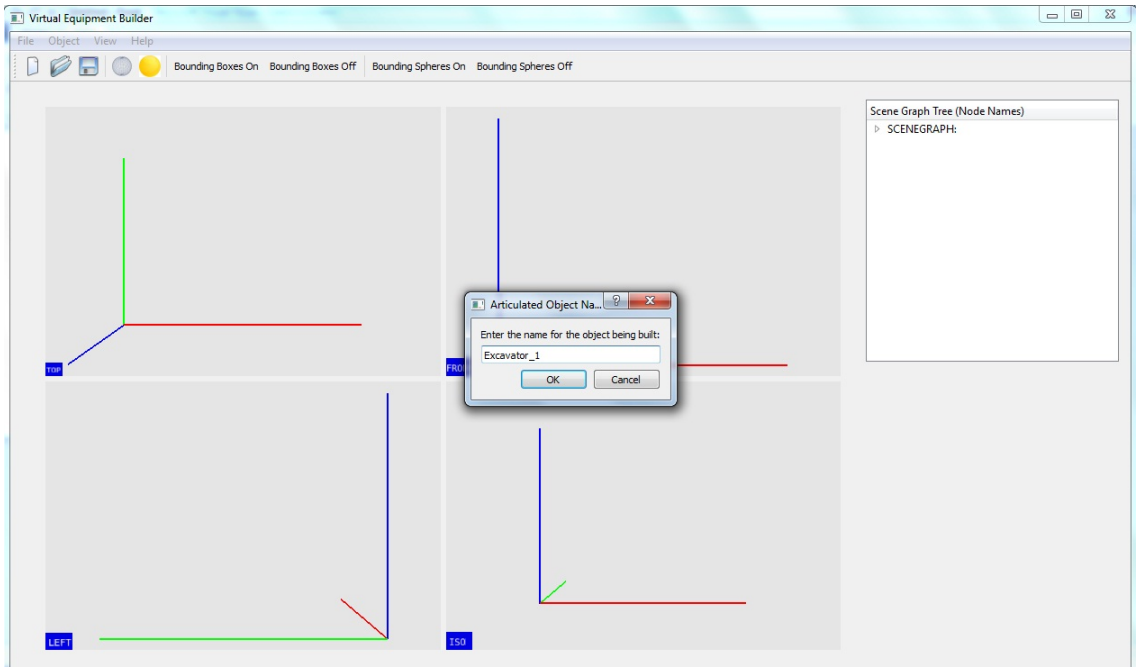

Fig. A.19: Equipment object name input dialog presented to the user

After an equipment object name is provided, the application presents a file open dialog box to the user. Through this input mechanism, the user is allowed to select the root or uppermost parent component of all other equipment components. Thus in the case of Excavator_1, the track component would be chosen and loaded at this stage, as it represents the root component being the parent of the cabin, boom, stick, and bucket. The file input dialog presented to the user is shown in Figure A.20.

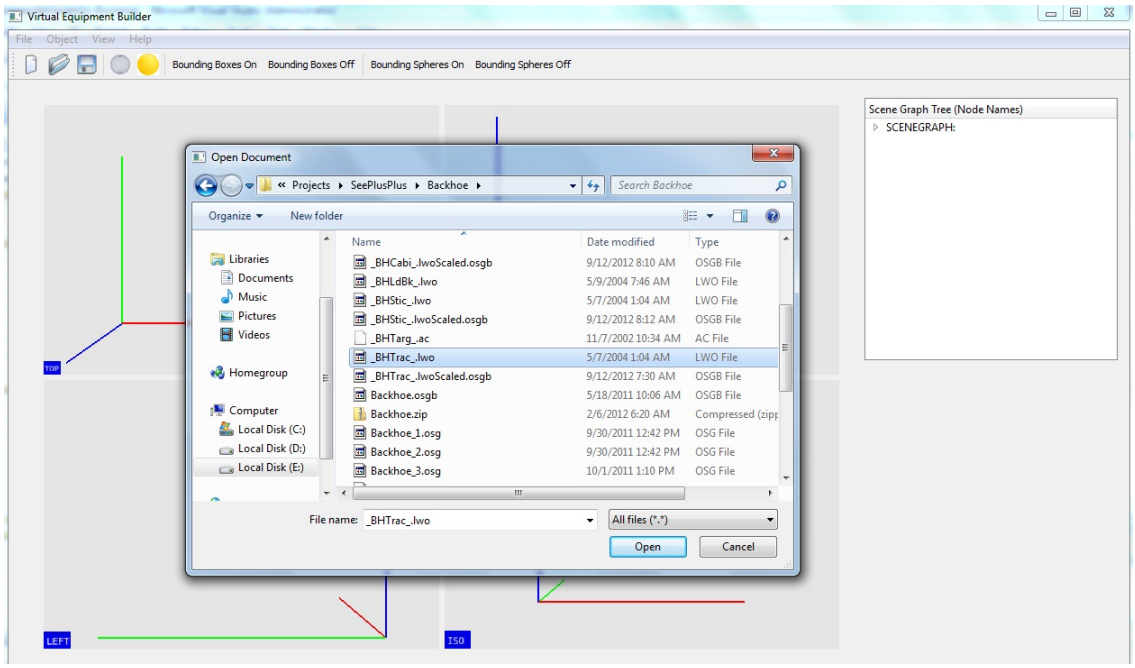

Fig. A.20: File input dialog for root or upper-most parent component

If the equipment component selected by the user is loaded correctly by the VEB application, a success message is displayed to the user. The views in the background display the equipment component that has been successfully loaded by the application. This implementation is shown in Figure A.21. VEB is implemented using OSG, thus any equipment component file type that is supported by native OSG can be loaded by the application. Additionally, the resized equipment components generated by the Dimensioner application can be used directly in VEB. A complete list of compatible 3D file formats can be found on the following web page –

http://www.openscenegraph.org/projects/osg/wiki/Support/UserGuides/Plugins

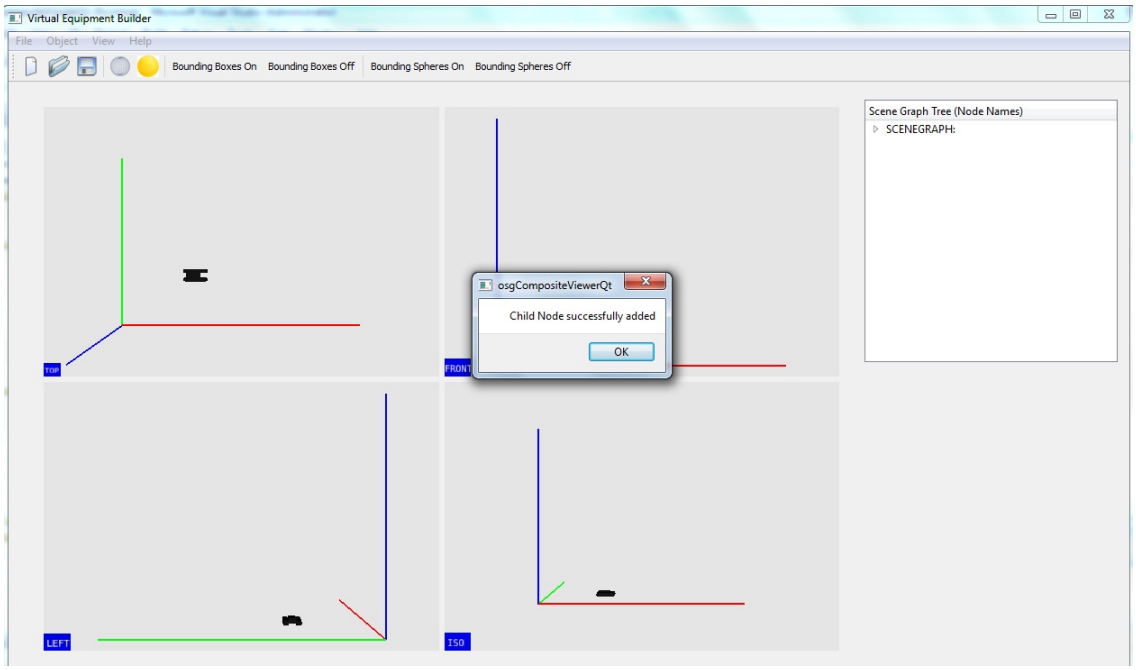

Fig. A.21: Success message displayed to user after root component model is loaded correctly

Once the track component has been successfully added to the application, a user requires a method to add a child node to it. This command is called ‘Add Child Node’ and is present under the Object menu of the application. Due to the use of scene graphs for implementing the parent-child hierarchy, the command uses the term ‘node’ as individual equipment components are internally referred to as a node in scene graph terminology. The terms node and component are thus used interchangeably.

The command to add subsequent nodes to the application is shown in Figure A.22. The figure shows two similar commands that are highlighted: Add Parent Node and Add Child Node. In the case of the uppermost parent node/component, a user would logically add the next component as its child node/component. However, for all other

nodes/components in the equipment's parent-child hierarchy, the application allows users to add a subsequent node/component as a parent or child of the existing chain.

Fig. A.22: Add Child Node/Add Parent Node commands under the Object menu to add parent/child nodes to existing 3D components

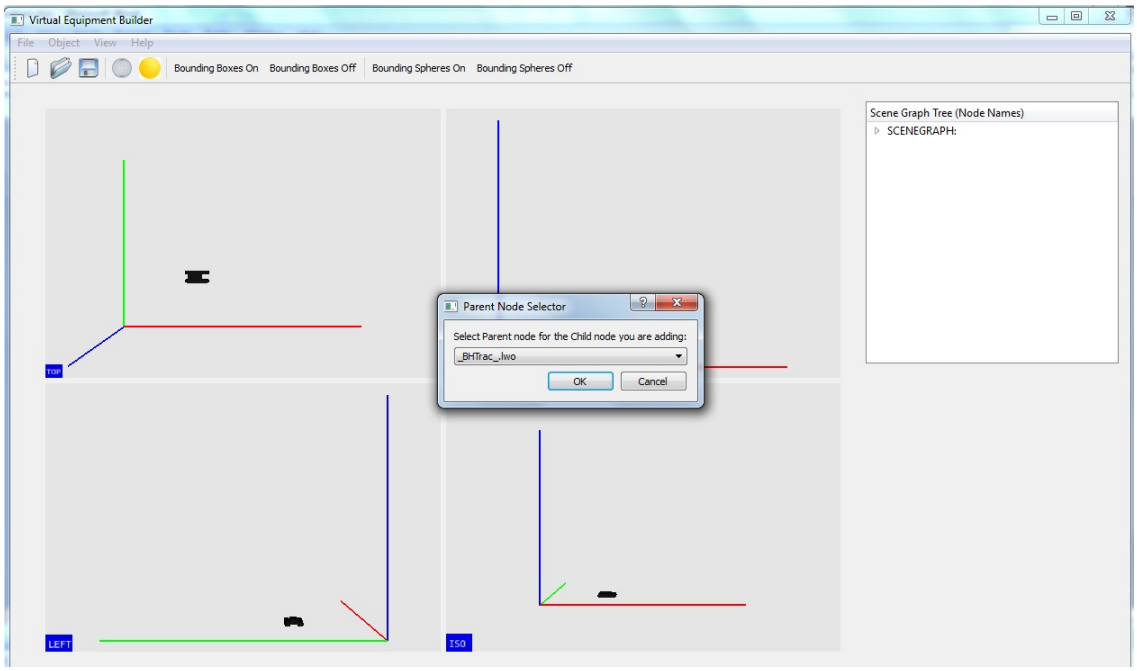

Fig. A.23: Dialog box for Parent node selection showing available nodes through a drop-down menu

Once the parent node selection is confirmed by the user, a file open dialog identical to that shown in Figure A.20 is presented to the user. A child component 3D model is then selected and loaded into the application, which if successful is indicated to the user through the message box, as shown in Figure A.21. Figure A.24 shows the VEB application after the cabin component is successfully added as a child node of the track component. However, as can be understood from Figure A.24, the cabin component's position with respect to its parent component (track) is not representative of the real world. Thus the newly added component needs to be translated and/or rotated to match the real- and virtual-world equipment.

Fig. A.24: VEB showing cabin component added as a child node of the track component

Users can translate/rotate components using the ‘Translate Node’ or ‘Rotate Node’ commands under the Object menu, as highlighted in Figure A.25. This command presents users with a drop-down menu, as shown in Figure A.26, that allows users to select the node they intend to rotate/translate. In the case of the equipment being built, the cabin component needs to be translated to match its real-world configuration. Thus confirming the selection of the cabin node presents users with a Translation-Rotation dialog box, as shown in Figure A.27.

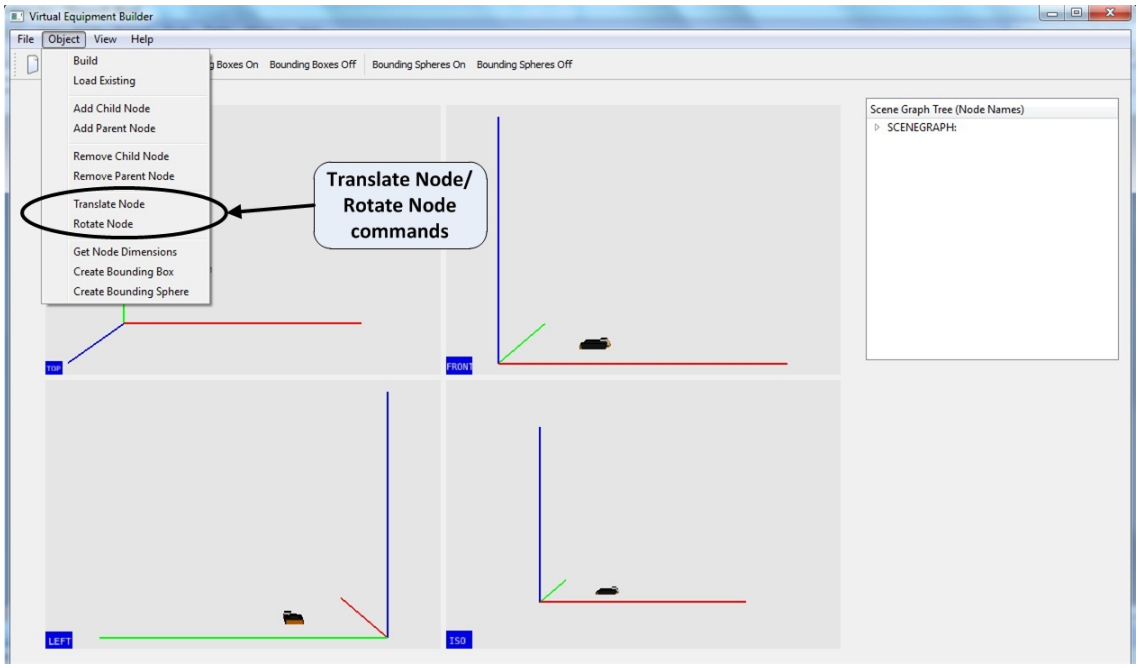

Fig. A.25: Translate Node and Rotate Node commands under the Object menu

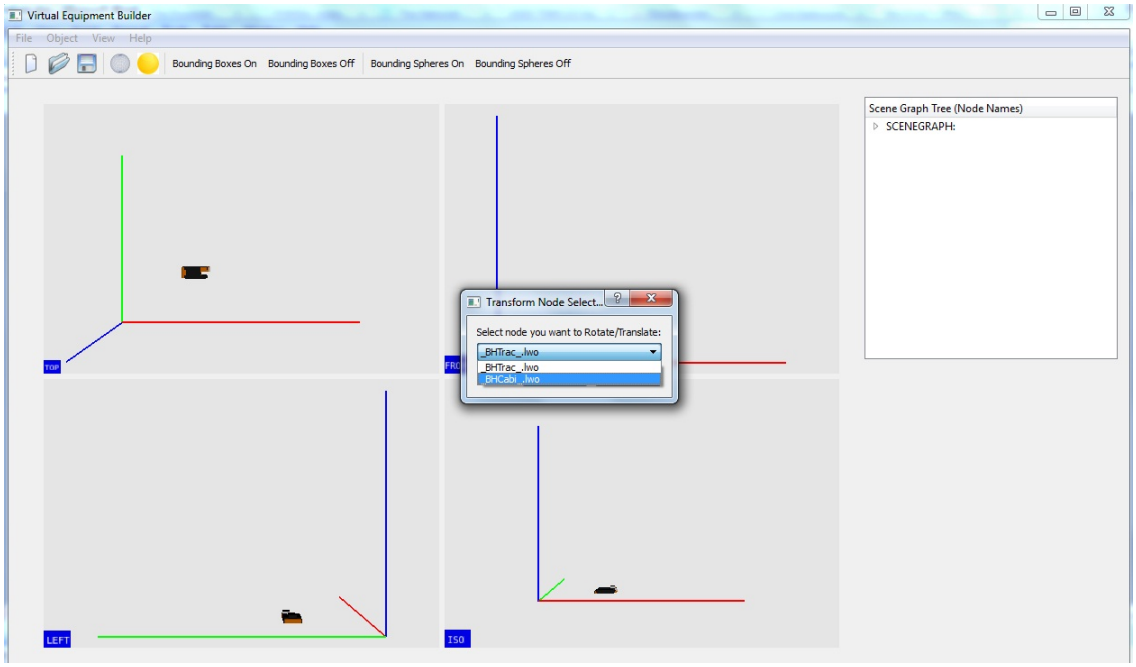

Fig. A.26: Drop-down menu for selecting node/component to rotate or translate

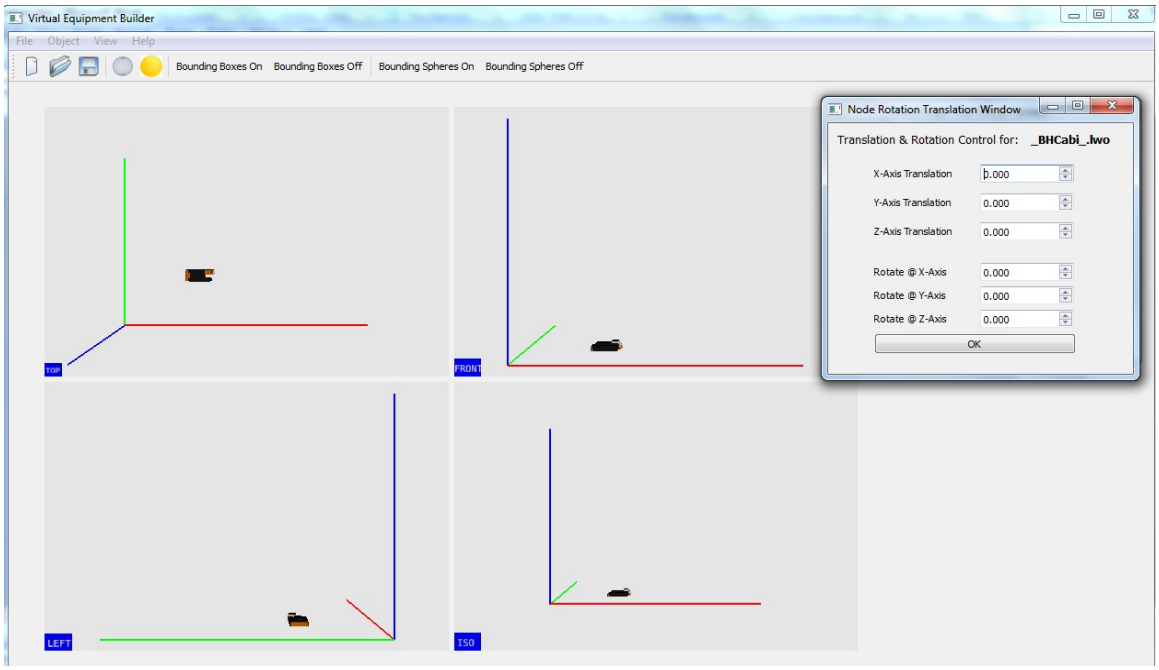

Fig. A.27: Translate/Rotate dialog for selected node

The same dialog box is presented to users for the Translate Node and Rotate Node commands. This is done to minimize the number of commands that users are required to initiate. The dialog provides users with six degrees of control (i.e., translation and rotation along X-, Y-, and Z-axes). After the component has been translated and/or rotated to the desired position, the 'Ok' command on the dialog confirms the positional changes. The cabin component, shown in Figure A.27, is translated upward along the Z-axis (represented by the vertical blue line) to match the configuration of the cabin-track hierarchy in the real and virtual worlds. The modified position of the cabin with respect to the track component as seen through the four views of the VEB is shown in Figure A.28.

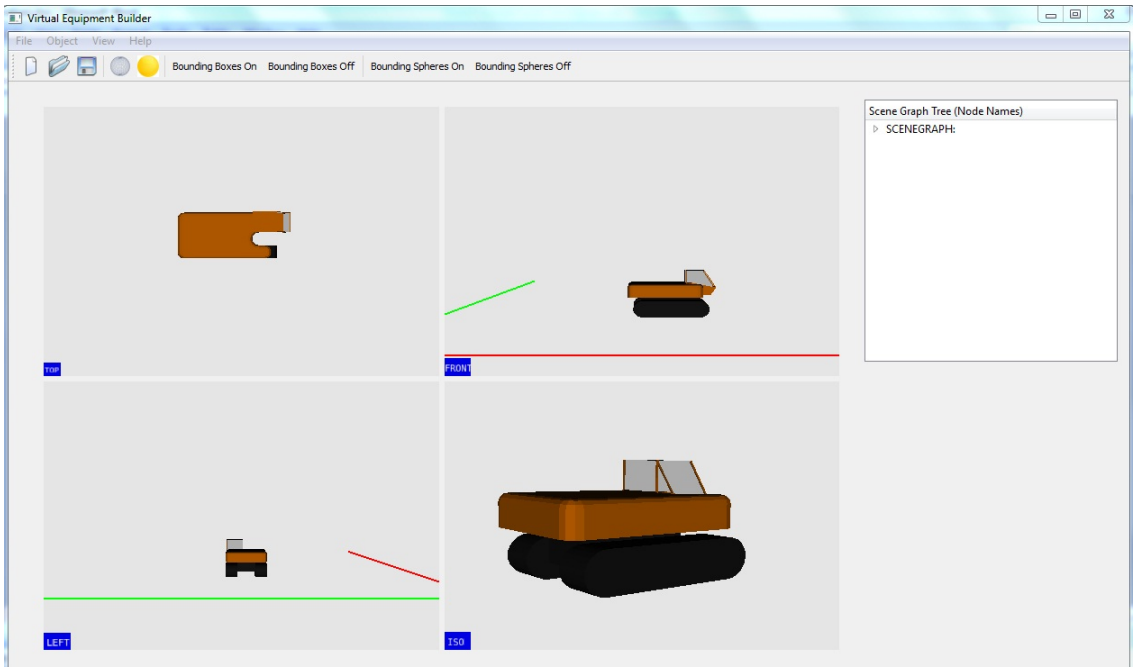

Fig. A.28: VEB showing the cabin component's modified position with respect to track

Once the child node has been repositioned to satisfy the user requirements, the next component can be added to the equipment hierarchy. In the case of the excavator, this component would be the boom. The subsequent components are added through a similar process as described in the preceding paragraphs—

1. Add Child Node
2. Select Parent Node from drop-down list
3. Load existing child component file
4. Success message displayed to user if loaded correctly
5. Rotate/Translate newly added node
6. Confirm positional changes made
7. Repeat steps 1–6 until all equipment components are loaded

In order to demonstrate steps 1 through 6 for the rest of the equipment components, screenshots from the actual process of building the 3D excavator are presented in Figures A.29 through A.33

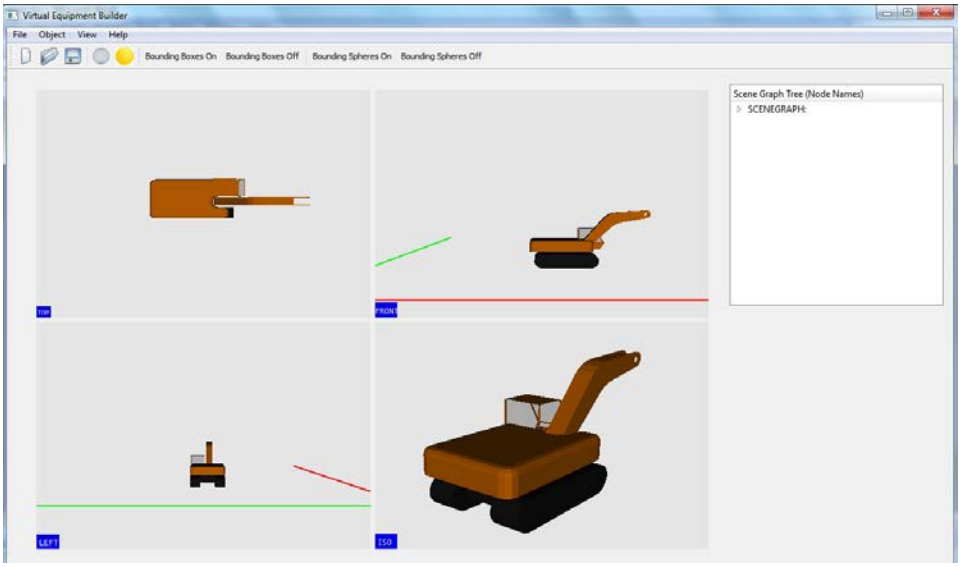

Fig. A.29: Boom component added to parent node (cabin), translated, and rotated to match user requirements

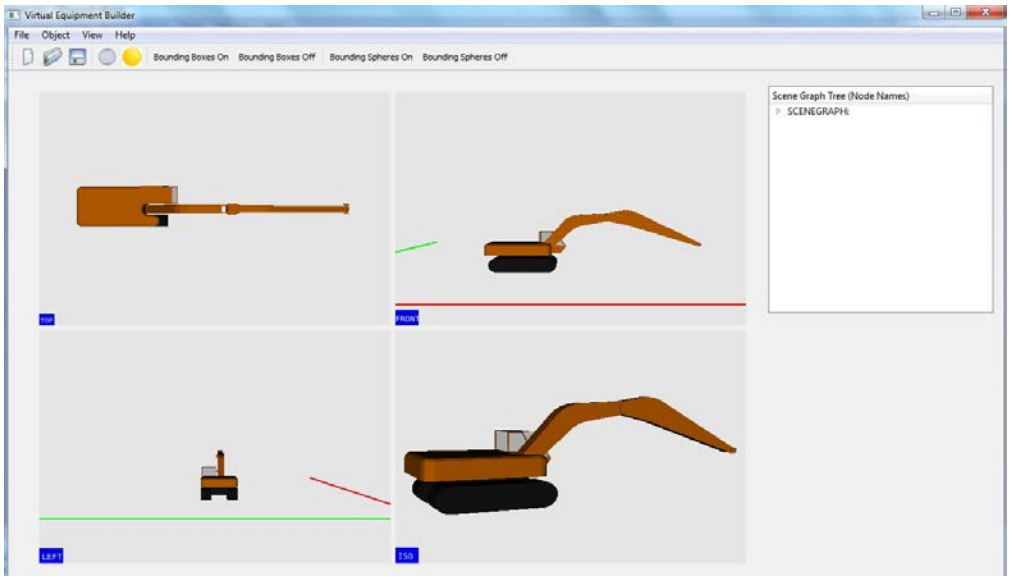

Fig. A.30: Stick component added to parent node (boom), rotated, and translated

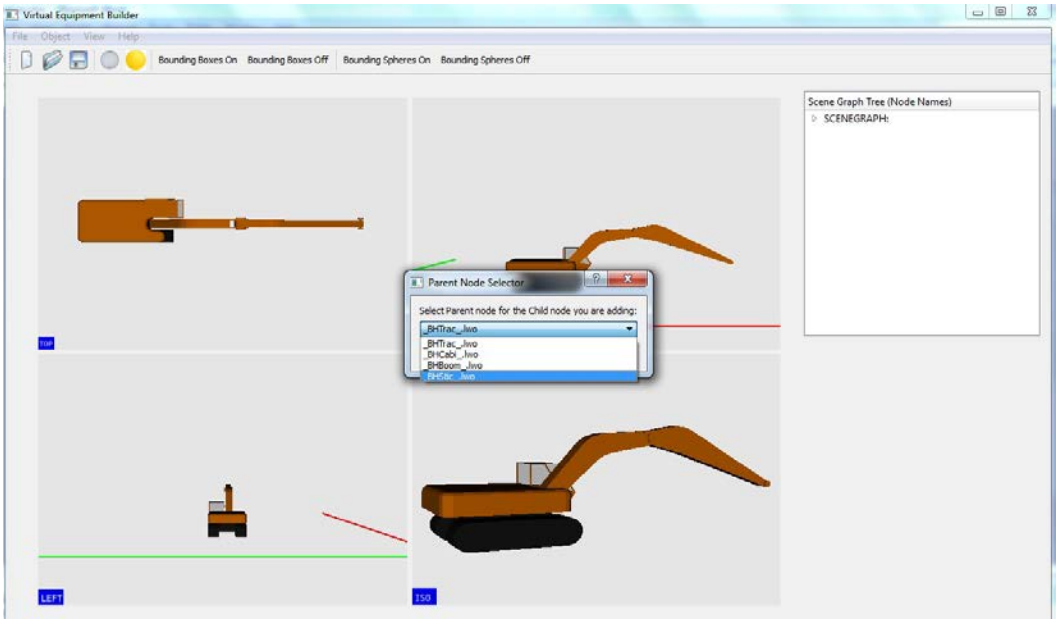

Fig. A.31: Parent node selection drop-down menu for adding the next component (i.e., the bucket component added to the stick component)

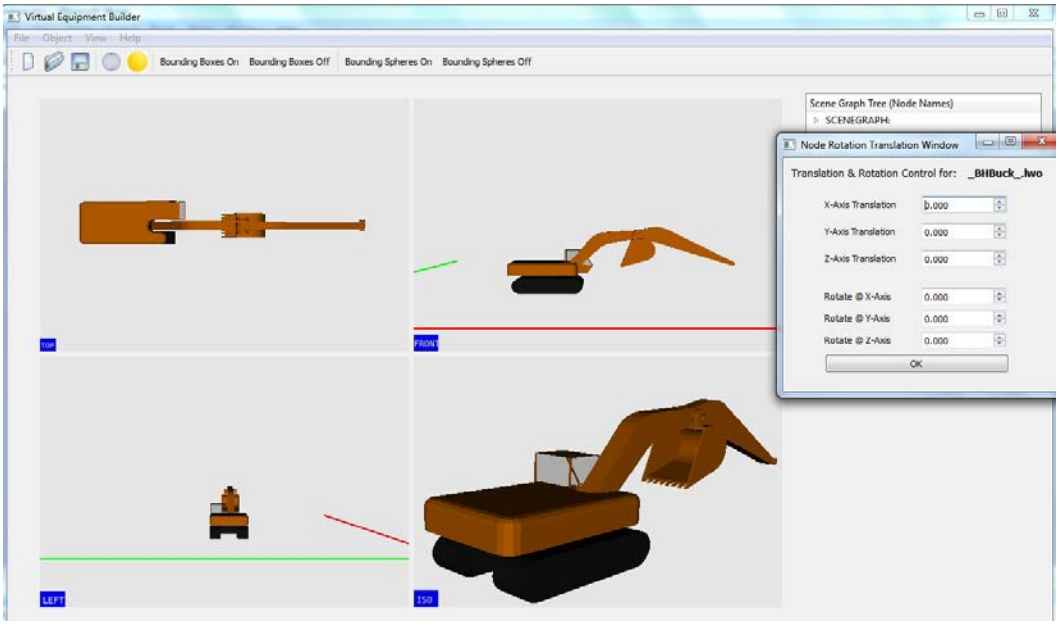

Fig. A.32: Bucket component added to equipment hierarchy but prior to undergoing any user-defined translations or rotations

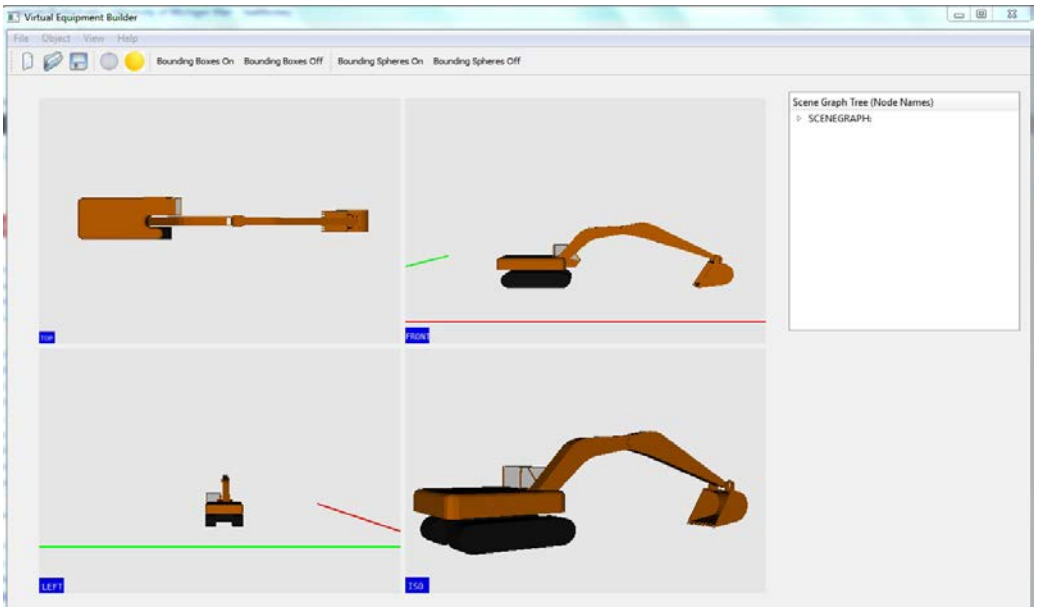

Fig. A.33: Bucket component rotated about Y-axis and translated along X-axis after being added to equipment hierarchy

After the bucket component is added to the rest of the equipment hierarchy, all of the components that make up the excavator are part of the equipment's parent-child hierarchy.

Thus after all of the components of the equipment have been added and repositioned to the user's requirements, the newly built equipment can be archived to a user-defined location to be re-used in 3D visualization applications such as SeePlusPlus. This action is initiated through the 'Save As' command under the File menu of VEB. The Save As command is highlighted in Figure A.34. Execution of the Save As command brings up the file save dialog, which allows users to save the 3D equipment model file to a user-defined location. This interface for archiving Excavator_1's file is shown in Figure A.35. A success message is displayed to the user if the file is correctly archived to the user's system, as shown in Figure A.36.

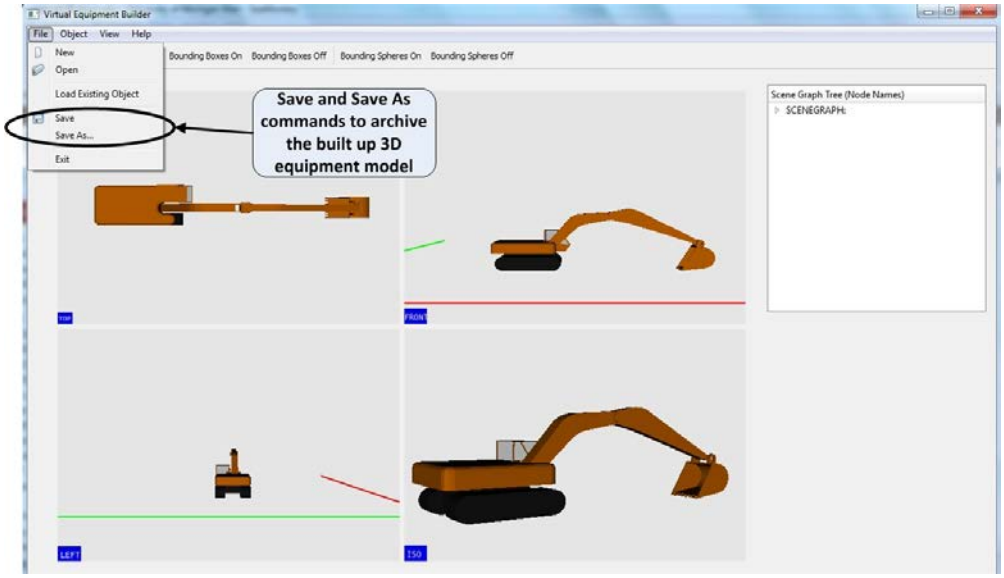

Fig. A.34: Save and Save As command to archive 3D equipment models is highlighted in the File menu

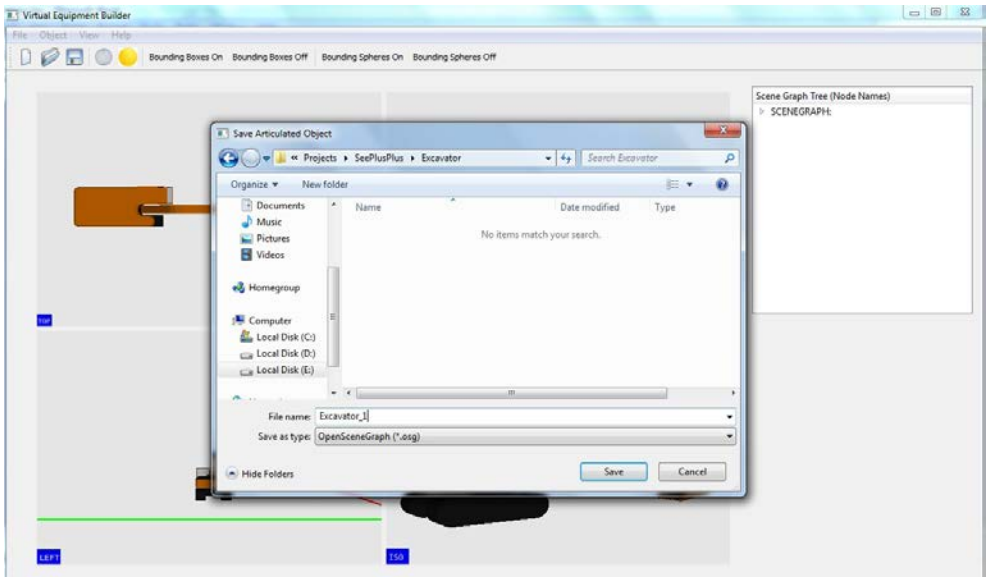

Fig. A.35: File save as dialog presented to the user after executing Save As command to allow user to define equipment file location

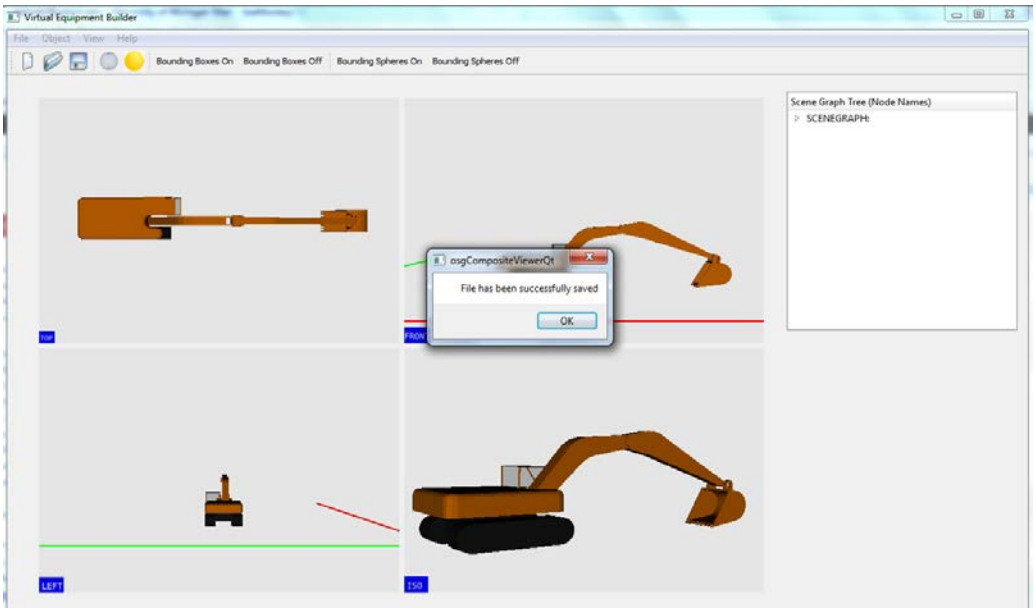

Fig. A.36: Success message displayed after 3D equipment file is correctly archived to the user's system

The VEB application allows users to create additional equipment models after they have finished creating and archiving the equipment that was being built. Thus the 'New' command icon, highlighted in Figure A.37, allows the VEB application to be restored to its startup state prior to loading any equipment component files. The New command is also present under the File menu of VEB. Figure A.38 shows the updated view after the New command is executed by the user. Once in this state, the VEB application can be used to create another equipment model by initiating the Build command or for loading existing 3D equipment models by executing the 'Load Existing Object' under the File menu.

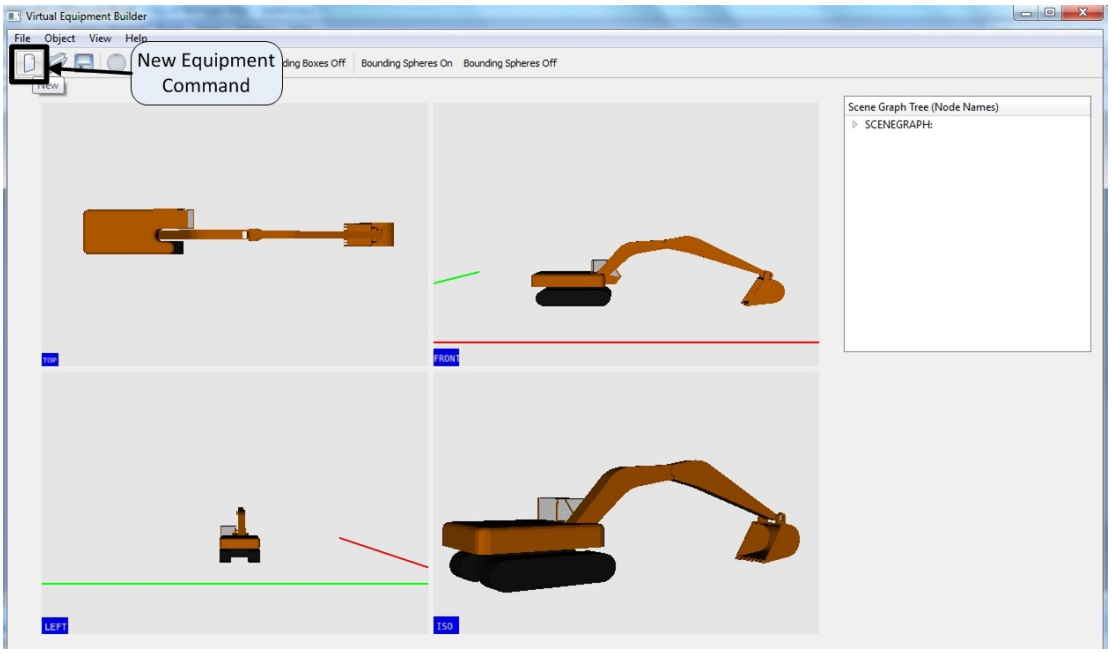

Fig. A.37: The 'New' command button to restore VEB to its startup state is highlighted

Fig A.38: The updated or restored state of VEB after execution of the 'New' command

The Load Existing Object is designed to allow users to reload existing 3D equipment objects into the VEB application. Through this command, users can check the validity and correctness of the 3D equipment objects that were built using VEB. This command is also designed to allow users to reload equipment objects and attach additional components to the already existing component hierarchy. The Load Existing Object is highlighted in Figure A.39. When the Load Existing Object command is executed, users are presented with the now-familiar file open dialog box, shown in Figure A.40. However unlike the Add Child command, the entire 3D equipment object will be loaded using the Load Existing Object command. This feature is demonstrated in Figure A.41 where the Excavator_1 equipment object that was created in the previous steps is now reloaded into the VEB application.

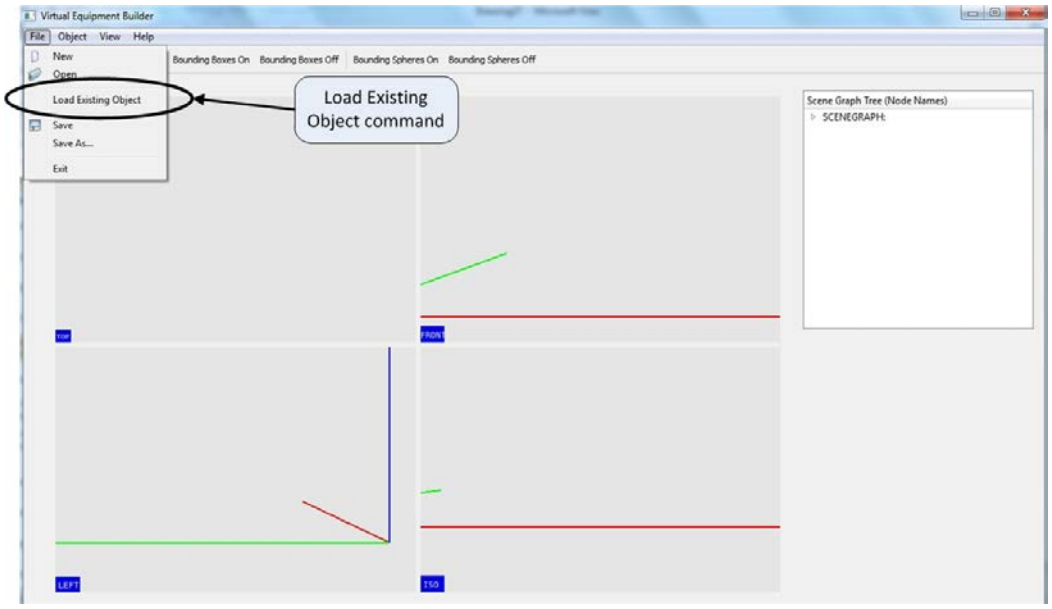

Fig. A.39: Command to load pre-existing 3D equipment file located under the File menu

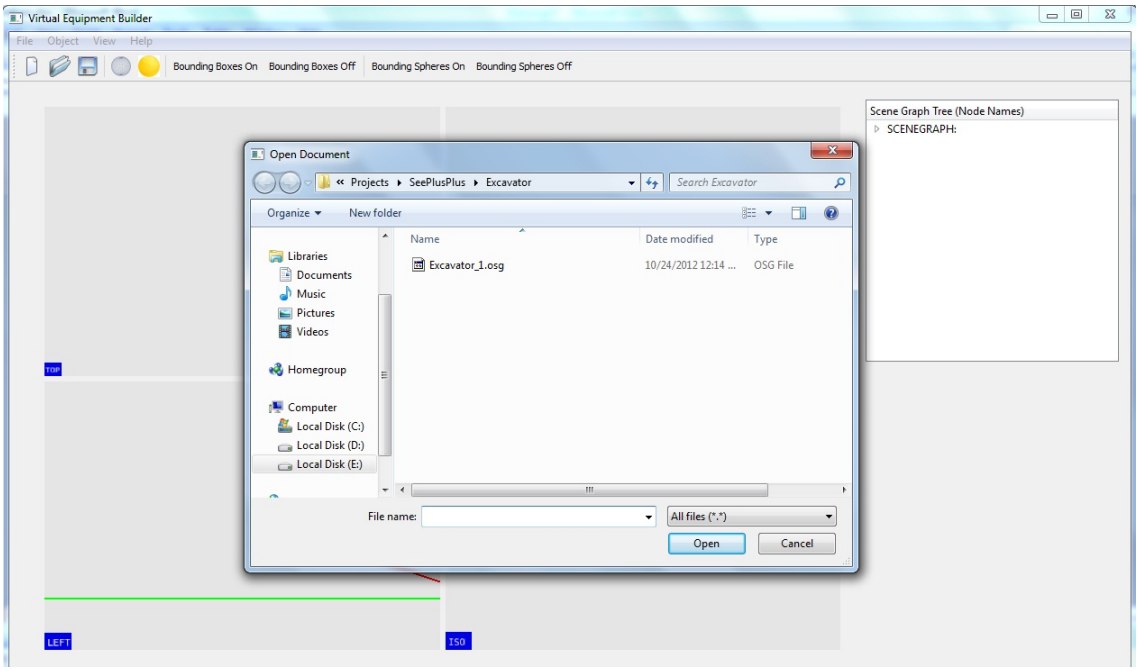

Fig. A.40: File open dialog to load existing equipment object file displayed to the user after execution of Load Existing Object command

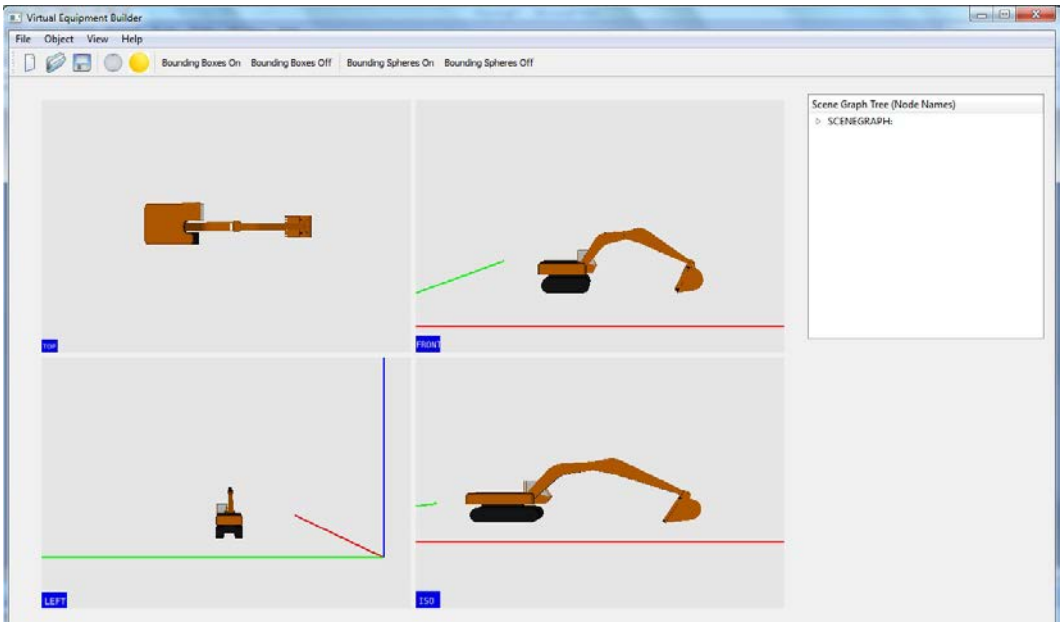

Fig. A.41: Pre-existing equipment object file correctly loaded and displayed to the user

The functions of the rest of the commands present in the VEB interface are presented in Figure A.42.

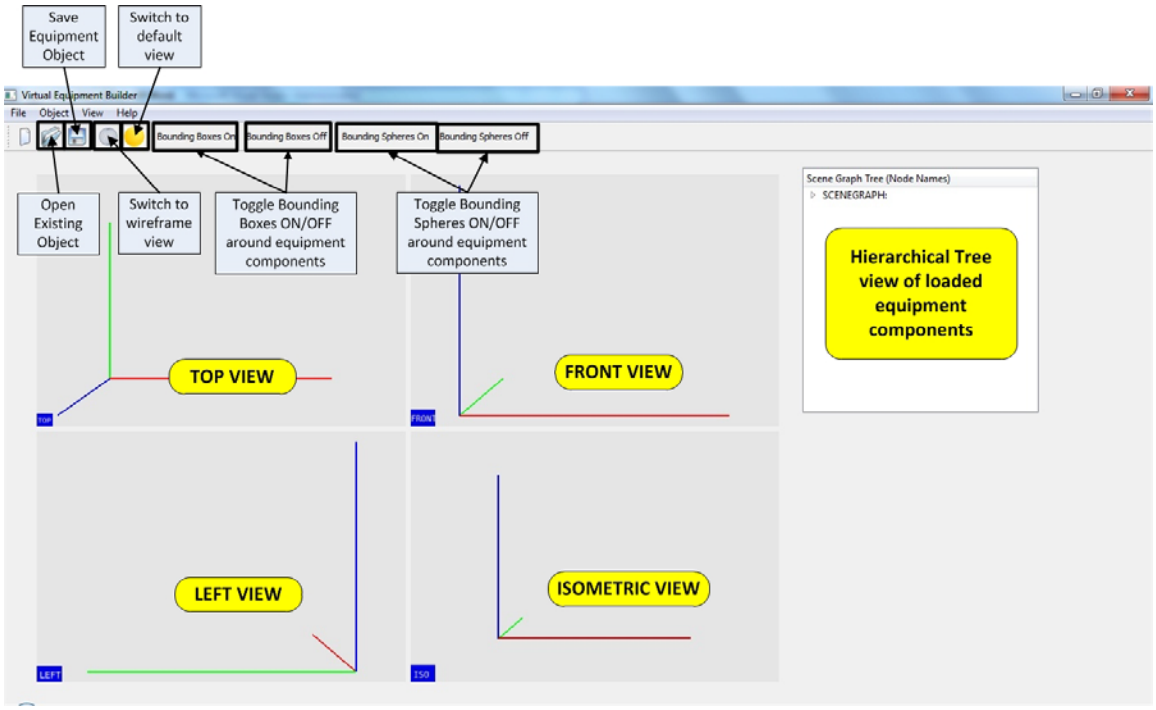

Fig. A.42: VEB interface with command functions highlighted and annotated

A.4 Geometric Primitive Description Files (TRIS Files)

An HVS is a completely virtual representation of a real-world operation. In this appendix, an HVS representing an excavation operation was looked at, and the building blocks required for creating it—the excavator, buried utilities, and terrain—were generated using a number of tools that were developed as an outcome of this research. However, an HVS is capable of analyses beyond those possible in the real world, for example proximity monitoring between the virtual entities present in an HVS providing distance and

collision information. A proximity query is defined between a pair of entities. Thus any proximity query requires two pieces of input: 1) the 3D models corresponding to the pair of entities involved, and 2) the position and orientation values of the pair of entities involved in the query.

The second piece of input concerning position and orientation values is provided using real-world sensor data streams. These sensor data streams are mapped to individual dynamic entities and their sub-components using the S2A2 module, the process for which is described in Appendix B. However, as part of the building blocks of an HVS, the creation of the geometry representation files for the pair of entities in a proximity query needs to be done at this stage.

The 3D models used in an HVS are polygon surface models (i.e., their surface is composed of a large number of geometric primitives such as triangles). A proximity query requires the list of primitives, the total number of primitives, as well as the vertices making up these primitives. The vertices of the geometric primitives need to be specified through their relative coordinates with respect to one another, for the base scenario where a 3D model has zero translating and zero rotation allocated to it (i.e., it is present at origin or (0, 0, 0) in virtual space). Thus such a file that describes the geometry of a 3D model through its constituent primitives for zero translation and rotation needs to be combined with the 3D model's global position and orientation in order to be used in proximity queries.

The geometric primitive description file needs to be created for every 3D model that is used in an HVS, and which will be part of proximity queries that a user will create within an HVS. The geometric primitives making up a 3D model's surface are all converted to triangle primitives through a process called tessellation. Thus, when viewed, the geometric primitive description file includes the number of triangle primitives along with their individual vertices' (x, y, z) locations. Thus the geometric primitive description files are given the file extension type '.tris' to signify the presence of triangle primitives in it.

The tris files for any 3D model can be generated through an application called 'TriangleCreator'. The code snippet below shows the output from a sample run of TriangleCreator where the boom component is provided as input. Option 1 is what would typically be used to ensure that '.tris' files are named appropriately.

```
>TriangleCreator.exe "E:\Projects\SeePlusPlus\Excavator\_BHBoom_.lwo"
Enter 1 if you want to save it by the PartName.
Enter 2 if you want to supply your own name:
1
The      file      name      we      are      writing      out      is:
E:\Projects\SeePlusPlus\Excavator\_BHBoom_.lwo.tris

The tris files can be found in the project folder: Hit any key to end
the program.
```

The following sample code shows the contents of a '.tris' file. As can be seen from the code sample below, the '.tris' file (representing the boom component from above) states the number of triangle primitives and then lists individual primitives through their (x, y, z) coordinates for each vertex. In the case of the boom 3D model, it is composed of 1024 geometric (triangle) primitives and each primitive is listed with the (x, y, z) coordinates

for its three vertices. TriangleCreator archives the '.tris' file in the same folder as the parent 3D model used to create it.

```
1024
6.33963 0.697992 0.858608
0.0236861 0.697992 0.330741
5.56905 0.697992 2.91987

0.0236861 0.697992 0.330741
6.33963 0.697992 0.858608
5.56905 0.697992 2.91987

11.3971 0.697992 0.322978
6.33963 0.697992 0.858608
5.56905 0.697992 2.91987

6.33963 0.697992 0.858608
11.3971 0.697992 0.322978
5.56905 0.697992 2.91987

-0.0672606 0.697992 0.322009
5.56905 0.697992 2.91987
0.0236861 0.697992 0.330741

5.56905 0.697992 2.91987
-0.0672606 0.697992 0.322009
0.0236861 0.697992 0.330741

.
.
.
.
.
.
.
.
.
.
.
```

In addition to the TriangleCreator application, both the B3M Creator and VEB applications are designed to generate '.tris' files for any 3D model that is created through these applications. It is also important to ensure a one-one correspondence between a 3D model and the '.tris' file associated with it. This is done through preserving the same name of the 3D model file and the '.tris' file it represents. Errors that result from

inconsistent naming conventions for ‘.tris’ files are described in Appendix B for specific modules of the SeePlusPlus application.

Appendix B

An introductory tutorial for using 'SeePlusPlus' - An application to create, save, and restore Hybrid Virtual Scenes

SeePlusPlus is a real-time 3D visualization application. It is based on the concept of a Hybrid Virtual Simulation (HVS) and uses the building blocks created in Appendix A. This appendix explains the procedure for using SeePlusPlus for creating an HVS, combined with its ability for proximity monitoring and usage of real-world sensor data. SeePlusPlus is designed as a graphical user interface whose screenshot is shown in Figure B.1. Each of the following sections in Appendix B describes the usage of a specific aspect of SeePlusPlus. As with Appendix A, the case of an excavation operation is used to demonstrate the workings of SeePlusPlus.

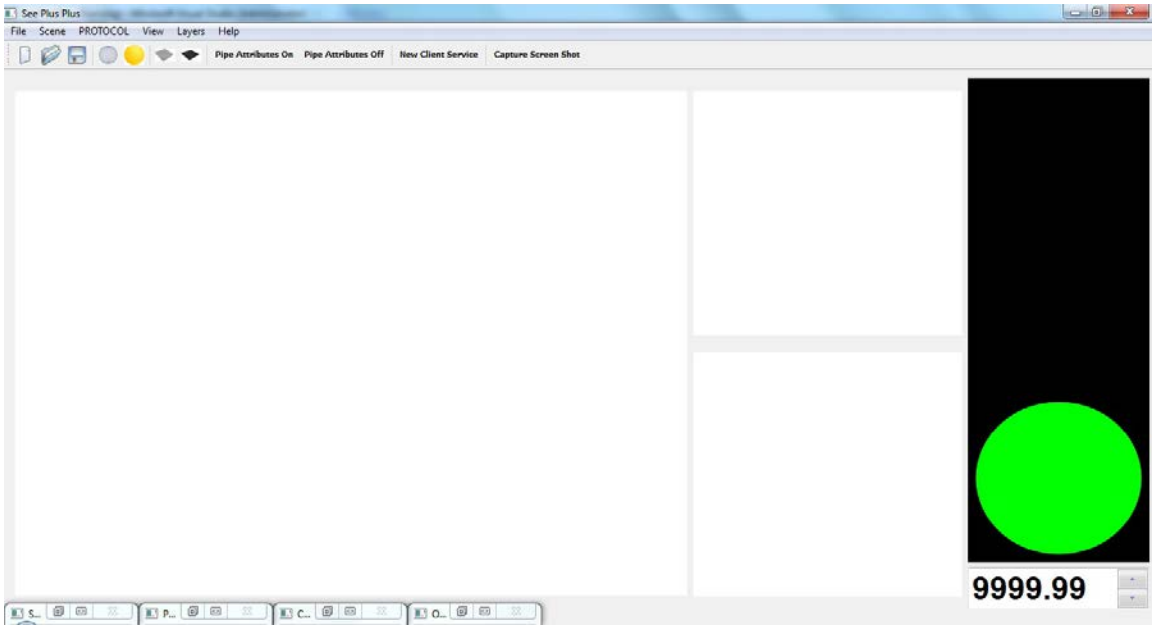

Fig. B.1: SeePlusPlus user interface at application startup

B.1 Overview of menus, modules, options, and affordances

SeePlusPlus is designed and implemented entirely as a graphical user interface, as shown in Figure B.2. The visual area of SeePlusPlus is divided into three sub-views: 1) a large main view that is user controlled and interactive in nature, 2) a fixed perspective view that provides a 3rd person view of the equipment involved in the operation, and 3) a fixed side-on view of the equipment involved in the operation. The right-most area of the interface has a visual warning widget that is designed to mimic a traffic light to warn users of impending danger. Details on interpreting the output from the traffic light widget can be found in section B.5. The bottom-right corner displays the distance between the equipment and the nearest object in its vicinity. This value displayed in this widget is not valid until proximity queries have been set up for the scene loaded in SeePlusPlus. Details on setting up proximity queries in SeePlusPlus are described in section B.3.

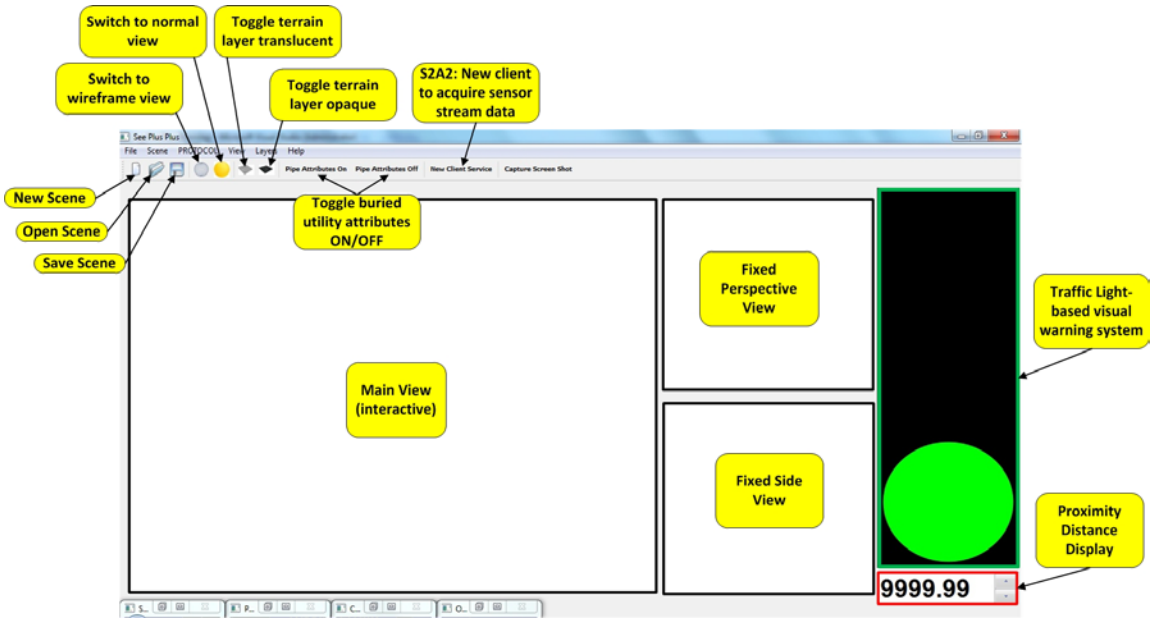

Fig. B.2: Interface elements of SeePlusPlus annotated

SeePlusPlus's extended functionalities—such as proximity querying and sensor data transfer—are implemented through independent modules, namely PROTOCOL and S2A2, respectively. These modules can be accessed by users through their separate interfaces and commands from the user menu. Details on setting up proximity monitoring queries through the PROTOCOL module are described in section B.3, and instructions on mapping real-world sensor data streams to 3D equipment components are presented in section B.4. The basic drop-down user menu for SeePlusPlus is shown in Figure B.3.

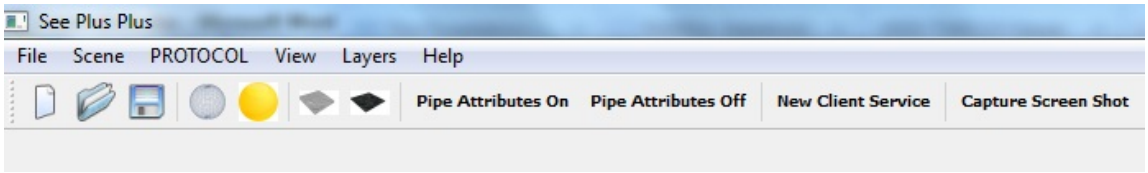

Fig. B.3: SeePlusPlus user menu

B.2 Creating an HVS scene in SeePlusPlus

This section describes the steps and data required to create an HVS scene in SeePlusPlus using building blocks from Appendix A. Once again, an excavation operation will be represented in the virtual world. The essential building blocks for an HVS representing an excavation operation are 1) the terrain, 2) the excavator (dynamic entity), and 3) the buried utilities (the static entity). Using the commands under the 'Scene' menu, as shown in Figure B.4, the following commands are executed by the user to create a HVS—

1. Scene → Load Terrain
2. Scene → Load Equipment
3. Scene → Load Utilities

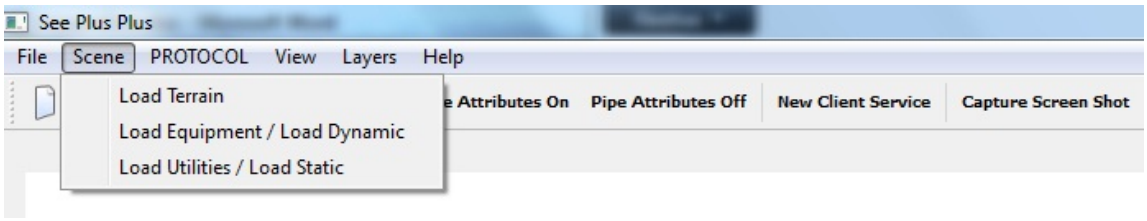

Fig. B.4: Commands placed under the Scene menu

The ‘Load Terrain’ command presents the user with a file open dialog. The user can navigate to the desired location and load the terrain model that was created in Appendix A, as shown in Figure B.5. If the file chosen by the user is successfully loaded, the main/interactive view of SeePlusPlus displays the terrain model, while the auxiliary side (fixed) views do not change their display until the equipment model is loaded. The updated SeePlusPlus interface after loading the terrain model is shown in Figure B.6.

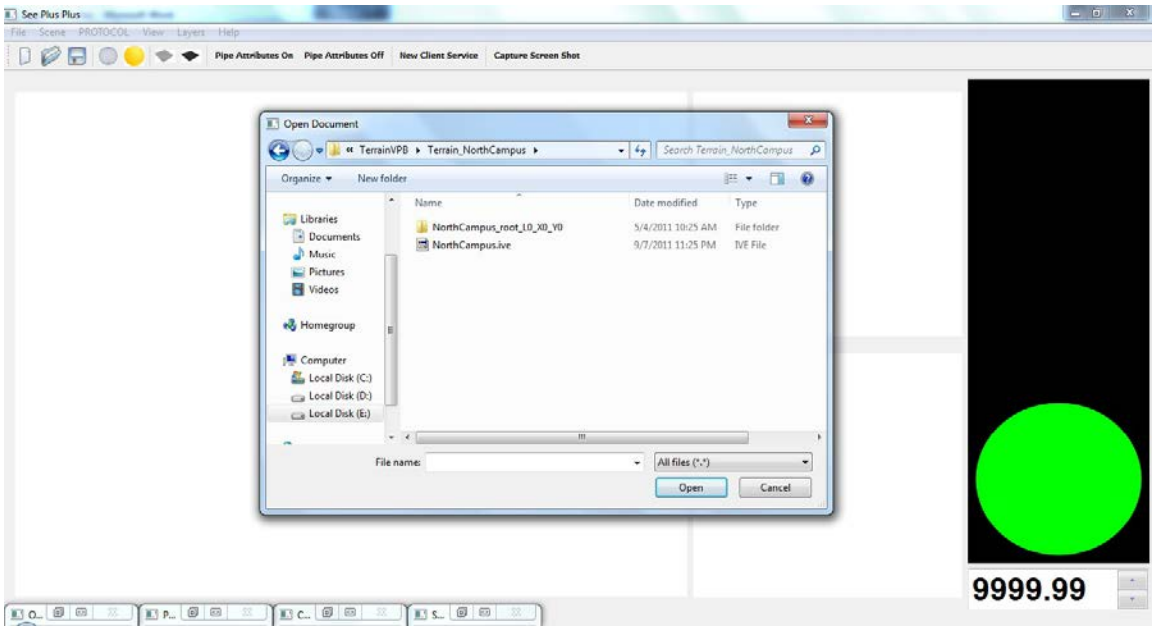

Fig. B.5: File open dialog to select terrain 3D model file

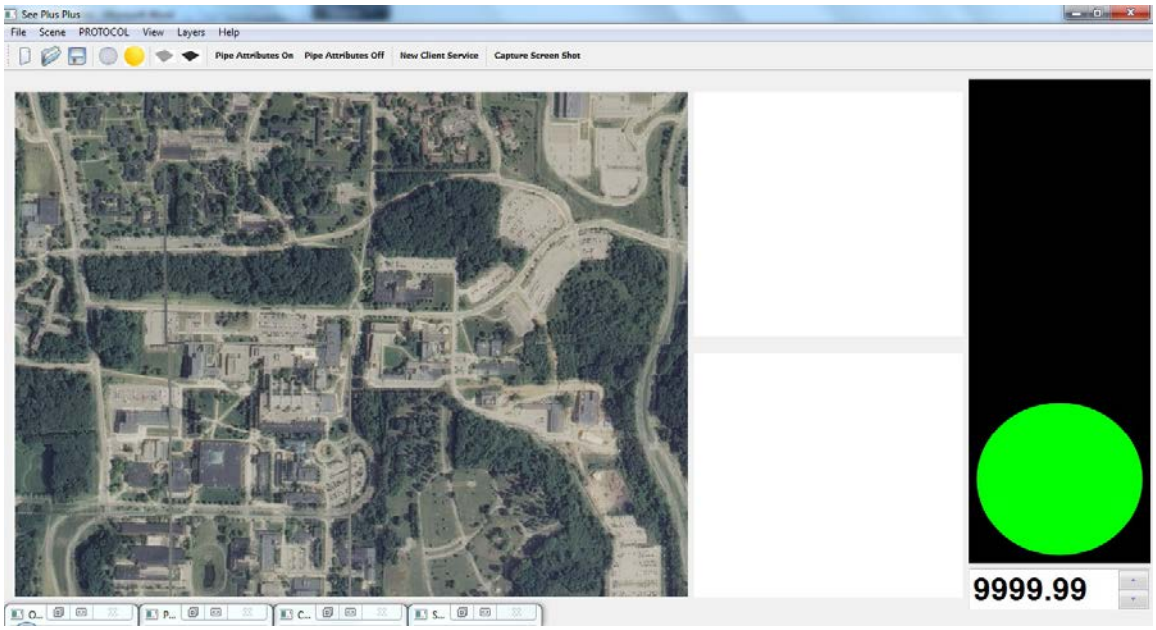

Fig. B.6: SeePlusPlus interface after terrain file has been successfully loaded

After the terrain model has been successfully loaded, the next step is the introduction of the 3D equipment model. It is important to point out that an equipment model cannot be loaded unless a valid terrain model exists in SeePlusPlus. This is done to ensure that the HVS is representative of the real world as much as possible. Execution of the 'Load Equipment' command presents the user with another file open dialog box. The 3D equipment model can be loaded by the user after navigating to the location where Excavator_1 from Appendix A was stored, as shown in Figure B.7. Alternatively, the user can also choose to load any other 3D equipment model that they may have created using the VEB application. If the 3D equipment object file is correctly loaded by the application, a coordinate input widget is presented to the user to obtain the location of the equipment in the real world in terms of latitude-longitude for placing the equipment accordingly in the virtual world. This input dialog box is shown in Figure B.8.

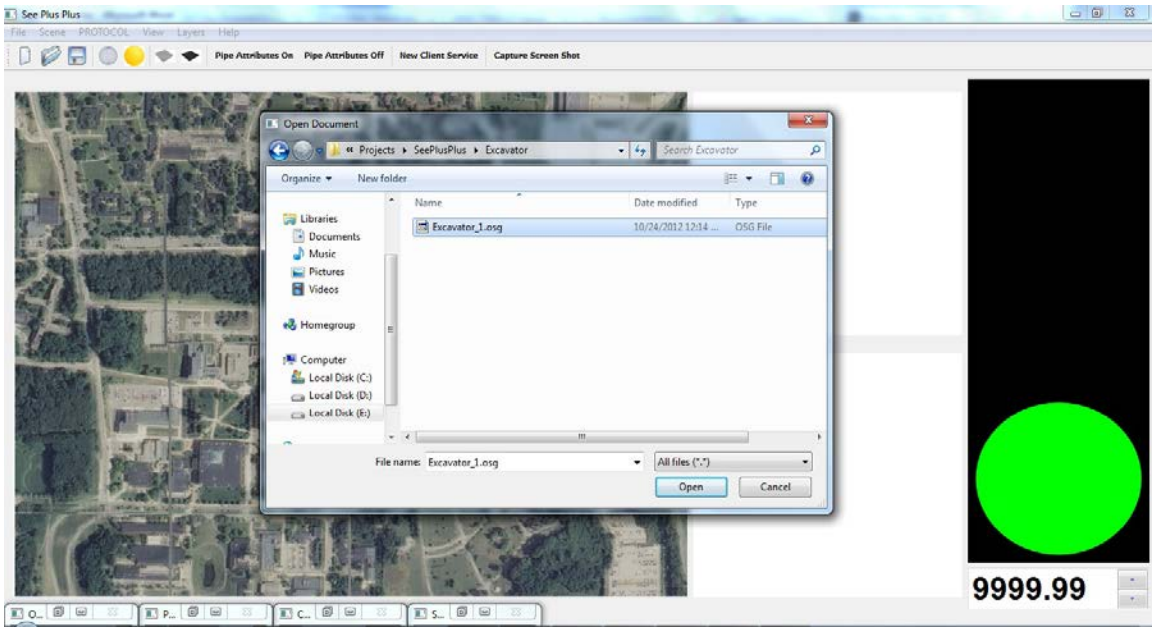

Fig. B.7: File open dialog for selecting 3D equipment object file

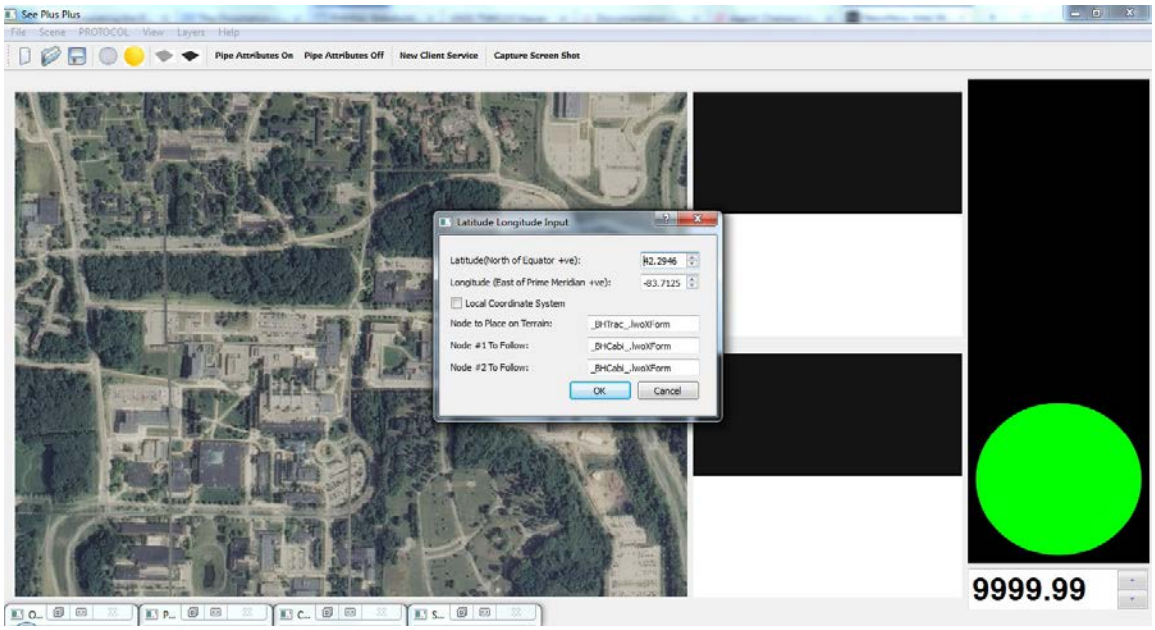

Fig. B.8: Coordinate input widget for placing equipment in the virtual world

The coordinate input widget is provided with a checkbox to specify if the coordinate system is local or global. An example of a local coordinate system is an indoor GPS

system where the location values provided in place of latitude-longitude are used directly without any conversion to a projected coordinate system. If the ‘Local Coordinate System’ checkbox is left unchecked, the latitude-longitude values provided by the user will be converted to their corresponding projected coordinate system values in order to coincide with the coordinate system of the terrain model and buried utility data. The coordinate input widget also provides an option for users to specify which component of the equipment the virtual cameras should follow in the fixed perspective and side views. After all input to the coordinate input widget is confirmed by the user, the equipment is successfully added to the HVS and is now shown in all three views provided by SeePlusPlus, as shown in Figure B.9. The cabin component is input as the component to be followed by the virtual cameras in SeePlusPlus.

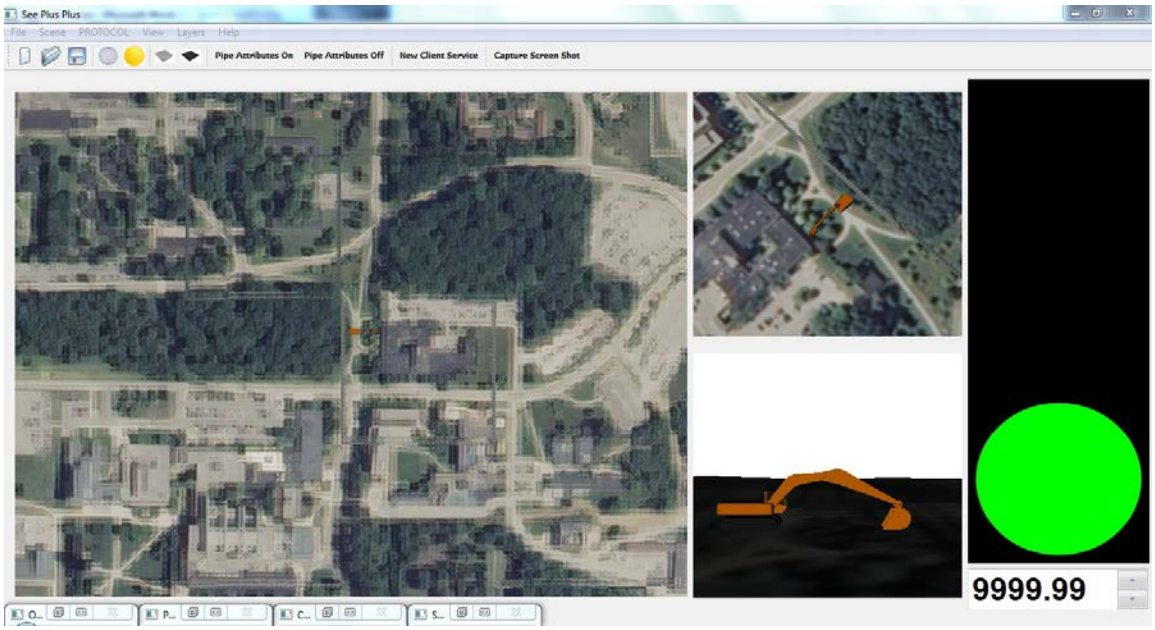

Fig. B.9: SeePlusPlus after the 3D equipment object is successfully introduced to the HVS

The final HVS building block to load is the buried utility data. The user executes the Load Utilities command under the Scene menu to bring up the file open dialog for 3D utility models, as shown in Figure B.10. In the case of the buried utility file open dialog, the user has an option to select multiple files at once, as it is possible for more than one utility model to be present at the jobsite location.

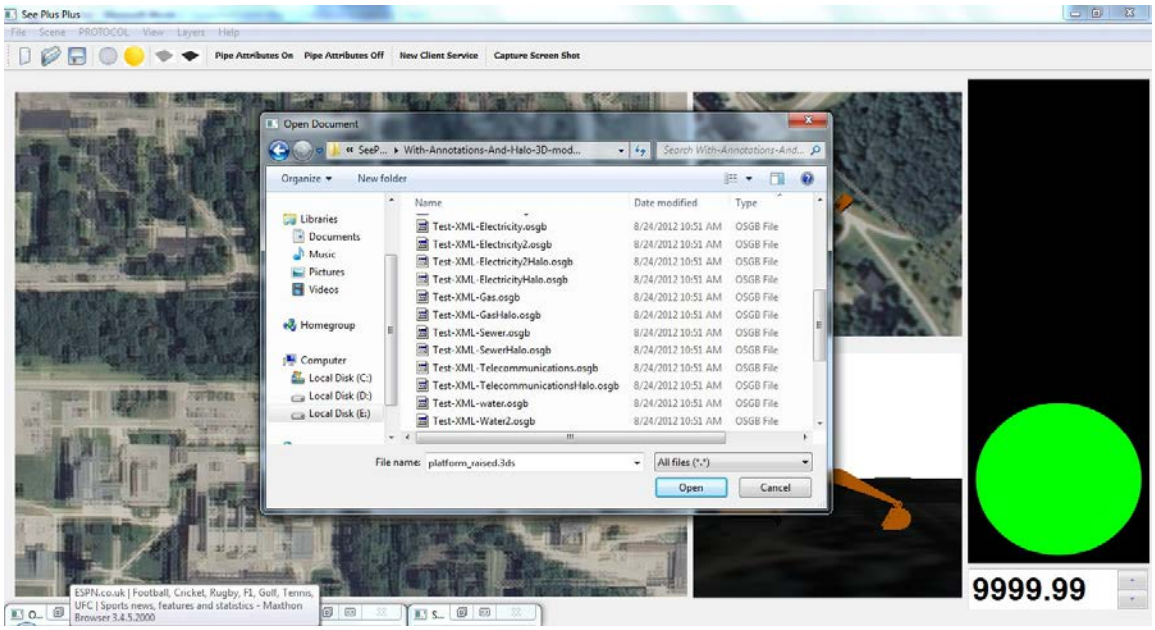

Fig. B.10: File open dialog for selecting single or multiple 3D buried utility files

The 3D utility models are added to the HVS if they are correctly loaded by the application. Any errors in loading are displayed to the user. However, even if the buried utility models are correctly loaded, they are not immediately visible as they lie below the terrain surface, which is opaque, and hence they remain visibly occluded, as seen in Figure B.11. In order to make the buried utilities visible, the transparency of the terrain needs to be altered from opaque to translucent, thus exposing the underlying 3D utility models, as shown in Figure B.12. The command to alter the terrain layer transparency is shown in Figure B.2.

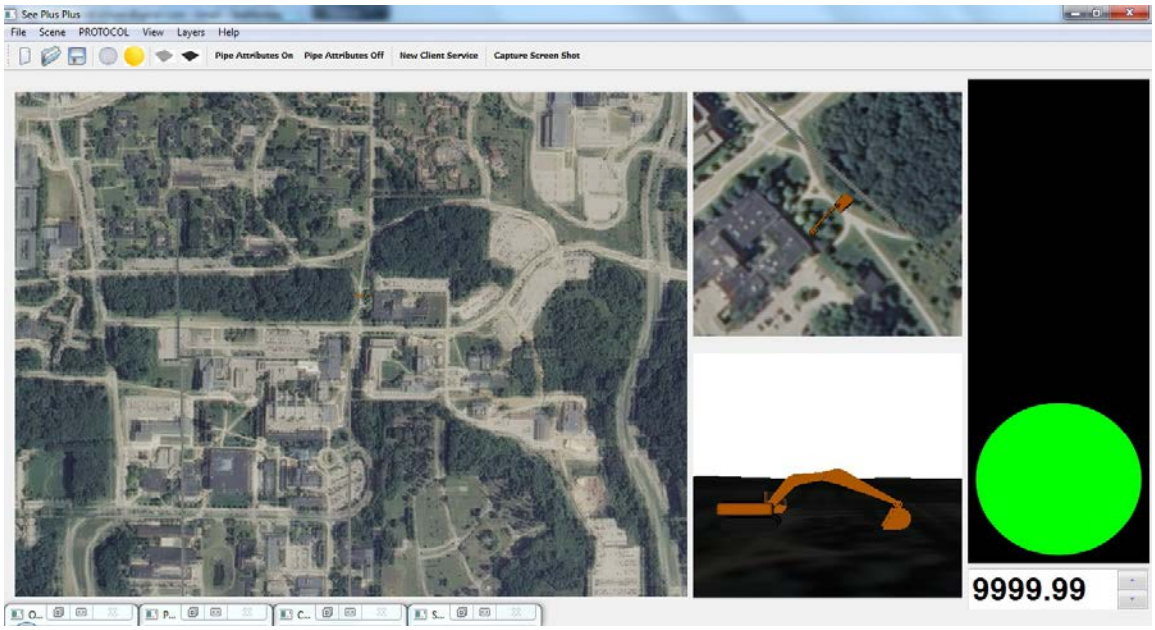

Fig. B.11: SeePlusPlus after the buried utility models are loaded into the application (utilities are below the opaque terrain surface)

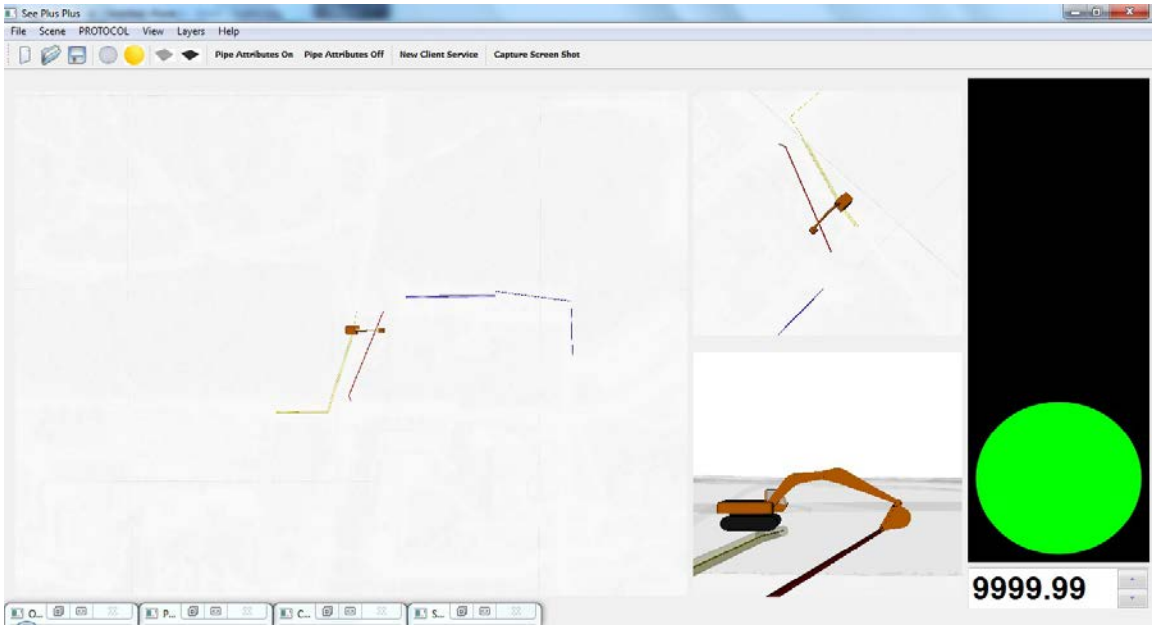

Fig. B.12: SeePlusPlus after the terrain layer transparency is altered from opaque to translucent using the command highlighted in Figure B.2

After the 3D utility models are loaded, the HVS scene is complete with respect to representation of all essential elements of a real-world excavation operation. It must be noted that the creation of an HVS on the jobsite at runtime is not always a feasible task. Hence SeePlusPlus allows the user to create an HVS beforehand and save it, and then reload it at runtime. The command to save an HVS to a user-defined location is present under the File menu, as shown in Figure B.13. When executed, the commands ‘Save Scene’ and ‘Save As…’ bring up a File Save dialog box, as shown in Figure B.14, which allows users to save an HVS scene file to their system. The file extension for HVS files is ‘.hvs’. SeePlusPlus informs the user that the current scene has been successfully saved to their system as a ‘.hvs’ file by displaying a success message, as shown in Figure B.15.

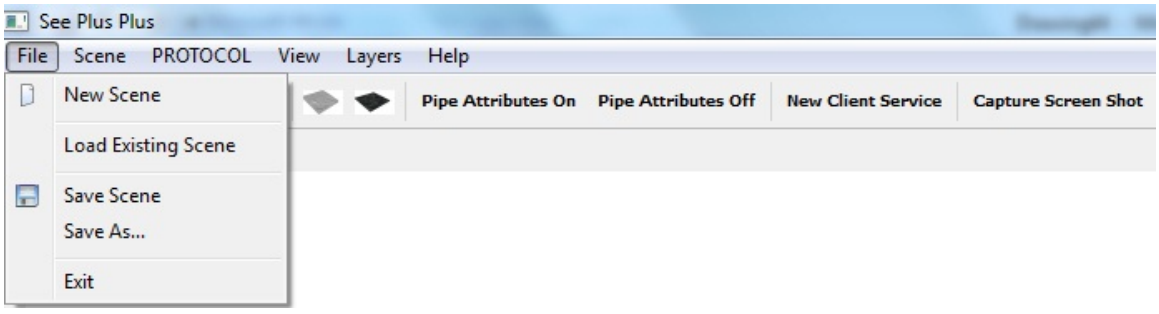

Fig. B.13: Save Scene and Save As commands to save an HVS file to a user-defined location

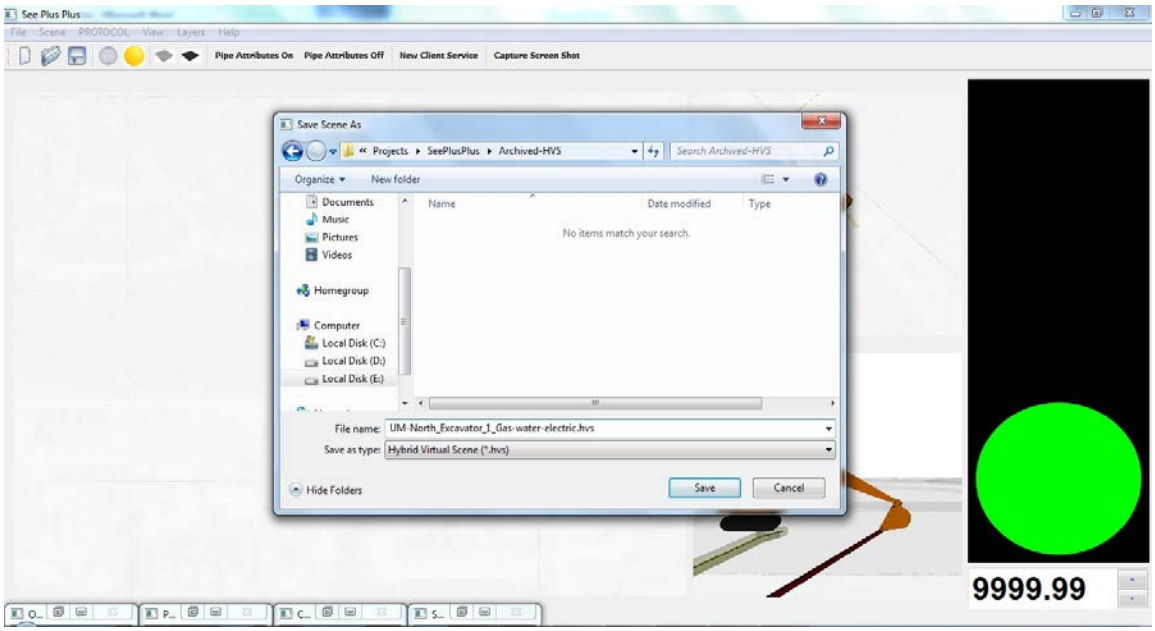

Fig. B.14: File Save As dialog box to save '.hvs' files to user-defined locations

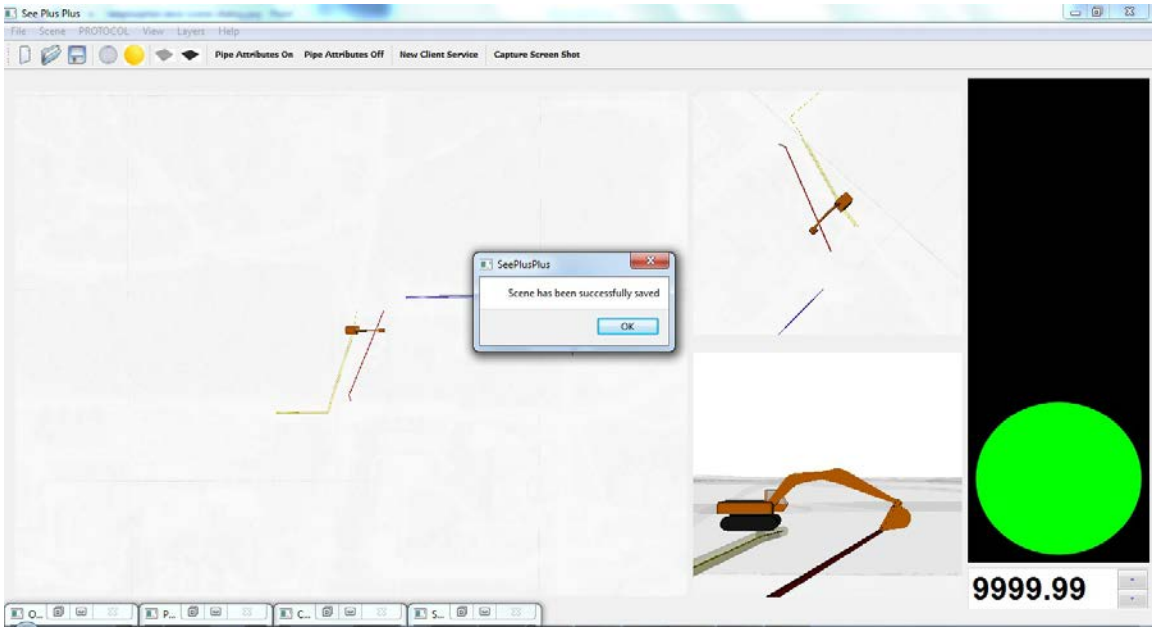

Fig. B.15: Success message displayed when HVS file successfully saved

During the process of creating an HVS by adding individual building blocks, it is possible for users to make mistakes or decide to add alternative 3D models than the ones

they used. The ability to undo the current scene creation and restart from step 1 is supported by SeePlusPlus. Users can restore the SeePlusPlus application to its startup state (i.e., no scene loaded in it by using the ‘New Scene’ command under the File menu or the New File action button), as shown in Figure B.2. Execution of the command results in the SeePlusPlus application state, as shown in Figure B.16.

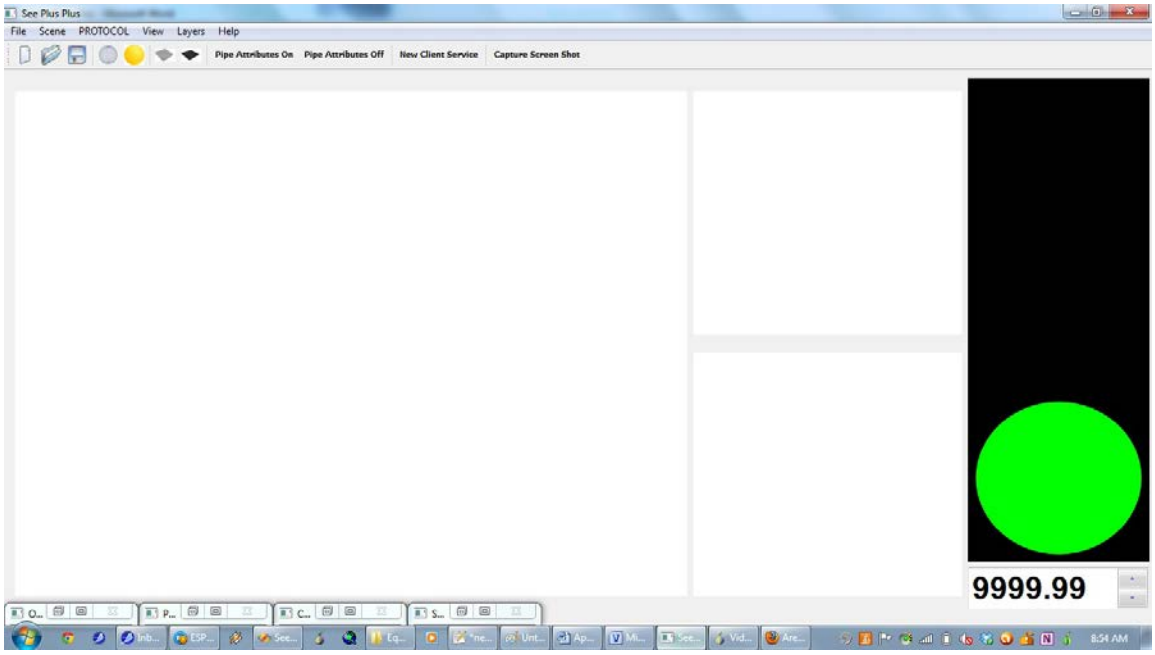

Fig. B.16: Updated views of SeePlusPlus interface after execution of ‘New Scene’ command

SeePlusPlus allows users to reload archived .hvs files using the ‘Load Existing Scene’ command located under the File menu, as shown in Figure B.17. Execution of this command presents the user with a File Open dialog, as shown in Figure B.18. If SeePlusPlus is successfully able to load the .hvs file, the output is as shown in Figure B.19, where the HVS scene created earlier in Appendix B is reloaded.

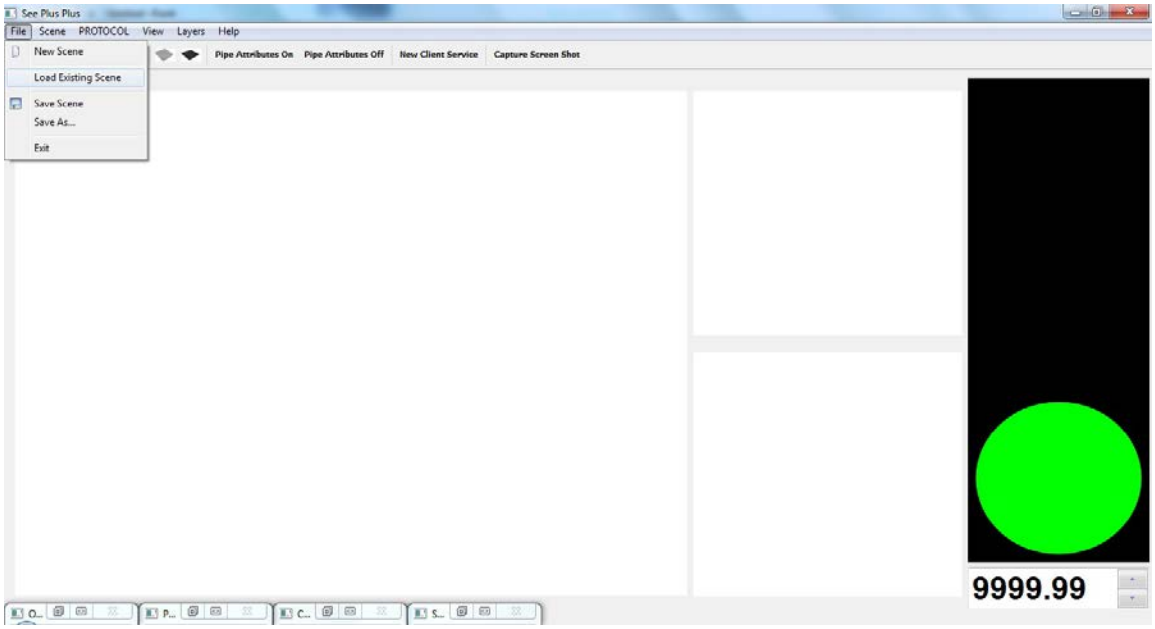

Fig. B.17: 'Load Existing Scene' command under File menu highlighted

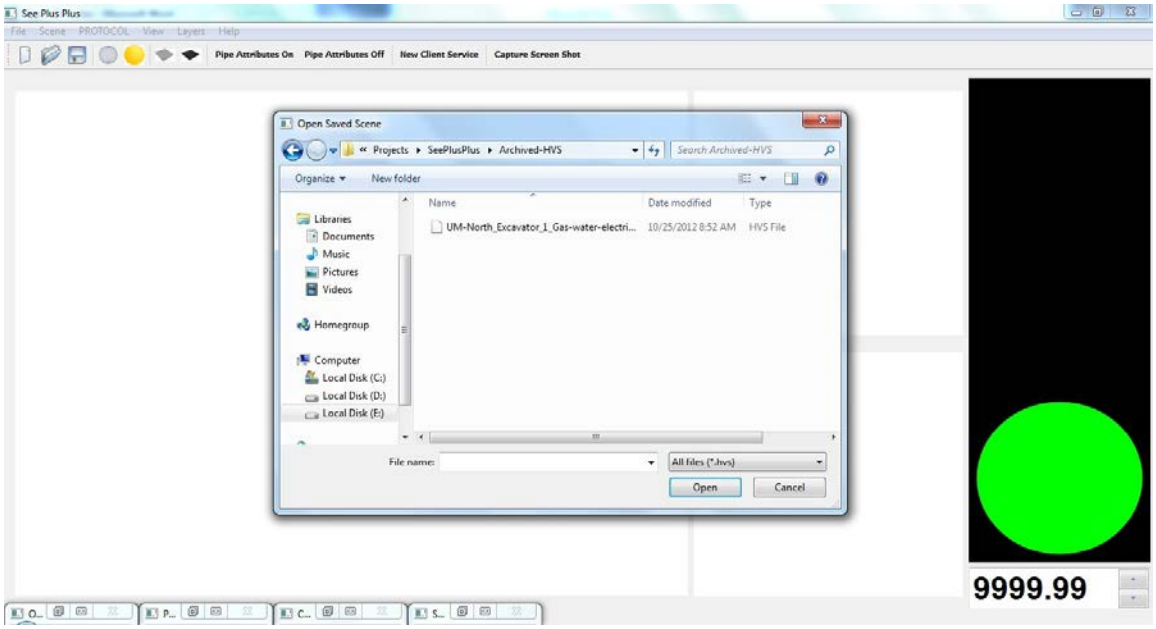

Fig. B.18: Open existing HVS scene dialog box

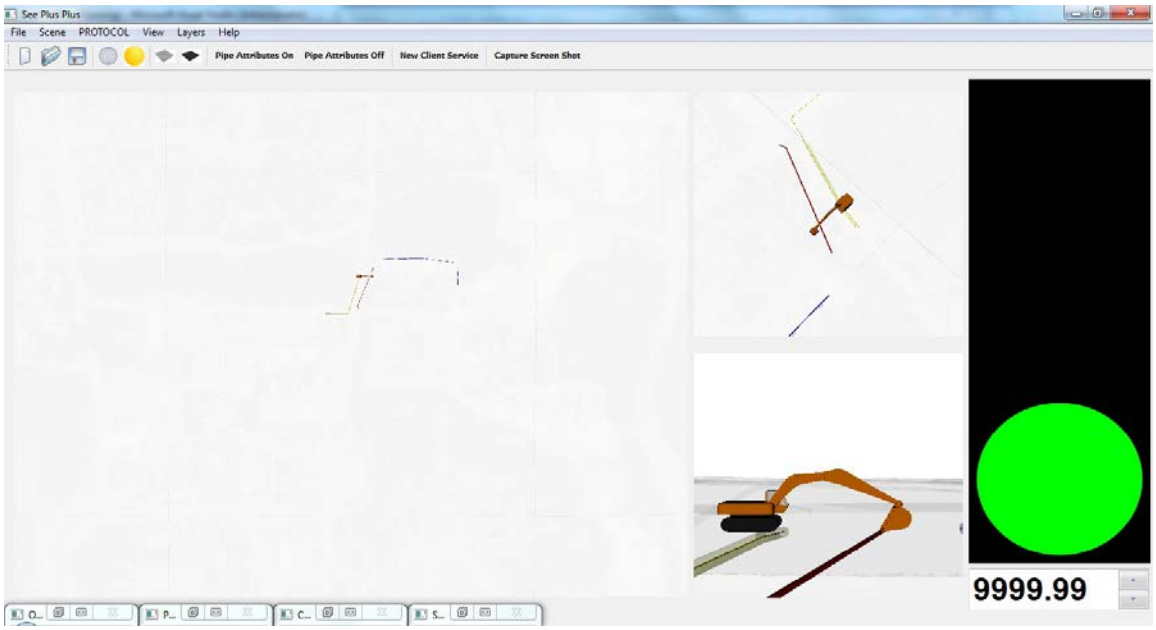

Fig. B.19: Existing scene successfully loaded into SeePlusPlus

B.3 Setting up PROTOCOL queries

In addition to providing real-time 3D visualization, SeePlusPlus also allows users to set up proximity queries between a pair of entities in the HVS scene. The module that is responsible for the setup, analysis, and archival of proximity queries is called PROTOCOL. Its name is derived from its ability to allow users to set up three types of queries: **Pro**ximity (distance), **To**lerance, and **Col**lision, thus resulting in the acronym PROTOCOL. The procedure for setting up proximity queries will be explained by using the HVS scene that has been used thus far in this appendix—a 3D terrain model, an excavator, and three buried utility models (water, electric, and gas). The PROTOCOL interface is minimized to the bottom-left corner of SeePlusPlus during application startup. As no entities are present in the HVS scene at startup, the PROTOCOL interface is also empty, as seen in Figure B.20.

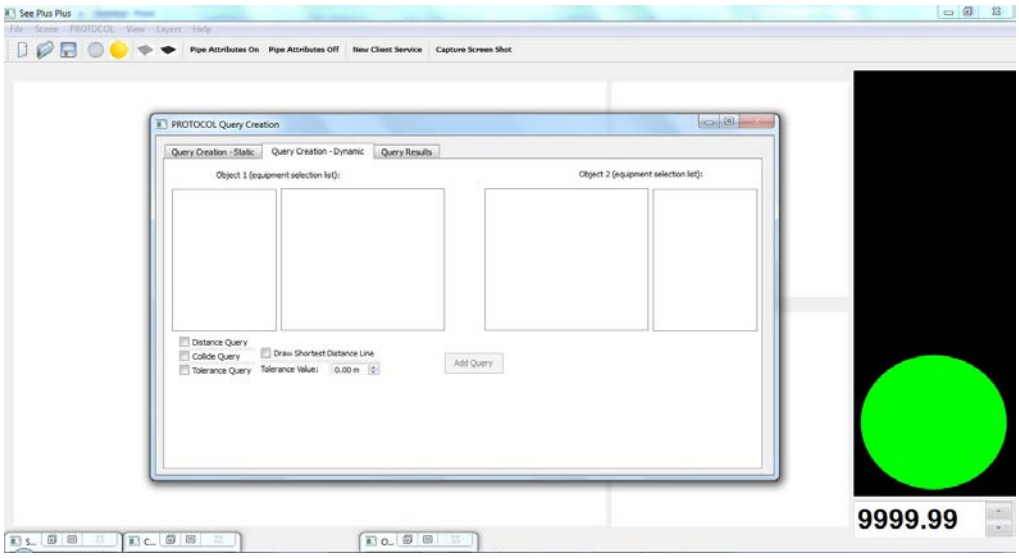

Fig. B.20: PROTOCOL widget at SeePlusPlus startup (i.e., empty HVS scene)

After the SeePlusPlus application is loaded with the HVS scene created earlier, the PROTOCOL interface is populated with both static and dynamic entities, as shown in Figure B.21.

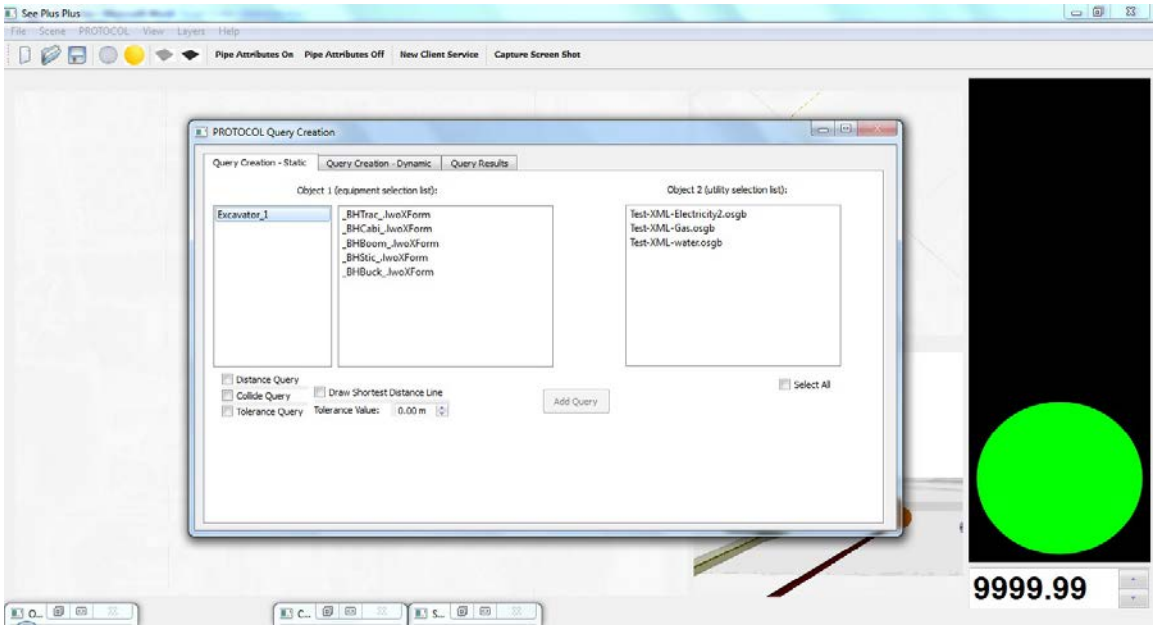

Fig. B.21: PROTOCOL interface showing static and dynamic entities available for setting up queries

The PROTOCOL interface is designed as a tabbed widget with three tabs: the first tab displays all dynamic entities (such as equipment components) on the left-side columns and all static entities (such as buried utilities) on the right-side column. A close-up view of the PROTOCOL static and dynamic query tabs is shown in Figure B.22.

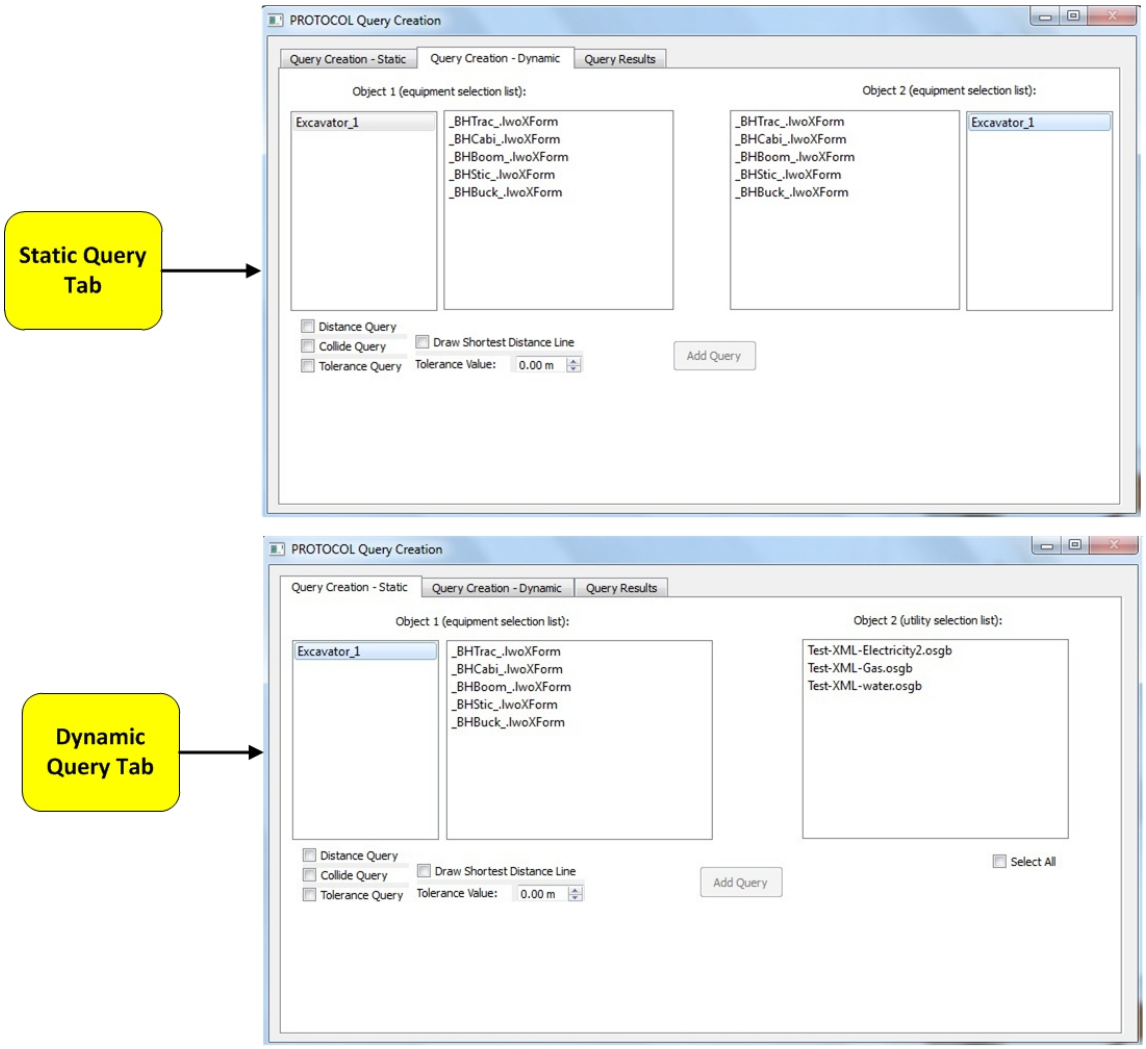

Fig. B.22: PROTOCOL static query and dynamic query tabbed widget

PROTOCOL queries are designed to be created through a one-one or one-many mapping. The left-most column in Figure B.23 is a list of all equipment objects in the HVS scene.

In the case of the scene being used in this appendix, there is only a single equipment object—Excavator_1. When a user clicks on an item in the equipment object list, the list alongside it (i.e., to its right, as seen in Figure B.23) displays all of the components making up the equipment object. Thus in the case of Excavator_1, the components making it up are _BHTrac_.lwoXForm, _BHCabi_.lwoXForm, _BHBoom_.lwoXForm, _BHStic_.lwoXForm, and _BHBuck_.lwoXForm. The right-most list (static object list) contains the list of buried utilities present in the scene.

In the case of dynamic queries, these would be made between a pair of components belonging to different equipment. The PROTOCOL interface will prevent erroneous queries, such as those between the same component on the same equipment, from getting created. An example of this would occur if the user selected the bucket component on the Equipment Component I and Equipment Component II lists. As with static queries, dynamic queries are also designed to be created using a one-one or one-many relationship, where a query can be generated between the bucket component of one excavator and the cabin, boom, and stick of a different excavator. In all other aspects, the rules for static and dynamic query creation remain the same. In this appendix, due to only a single excavator being present, a demonstration of dynamic query creation is not provided.

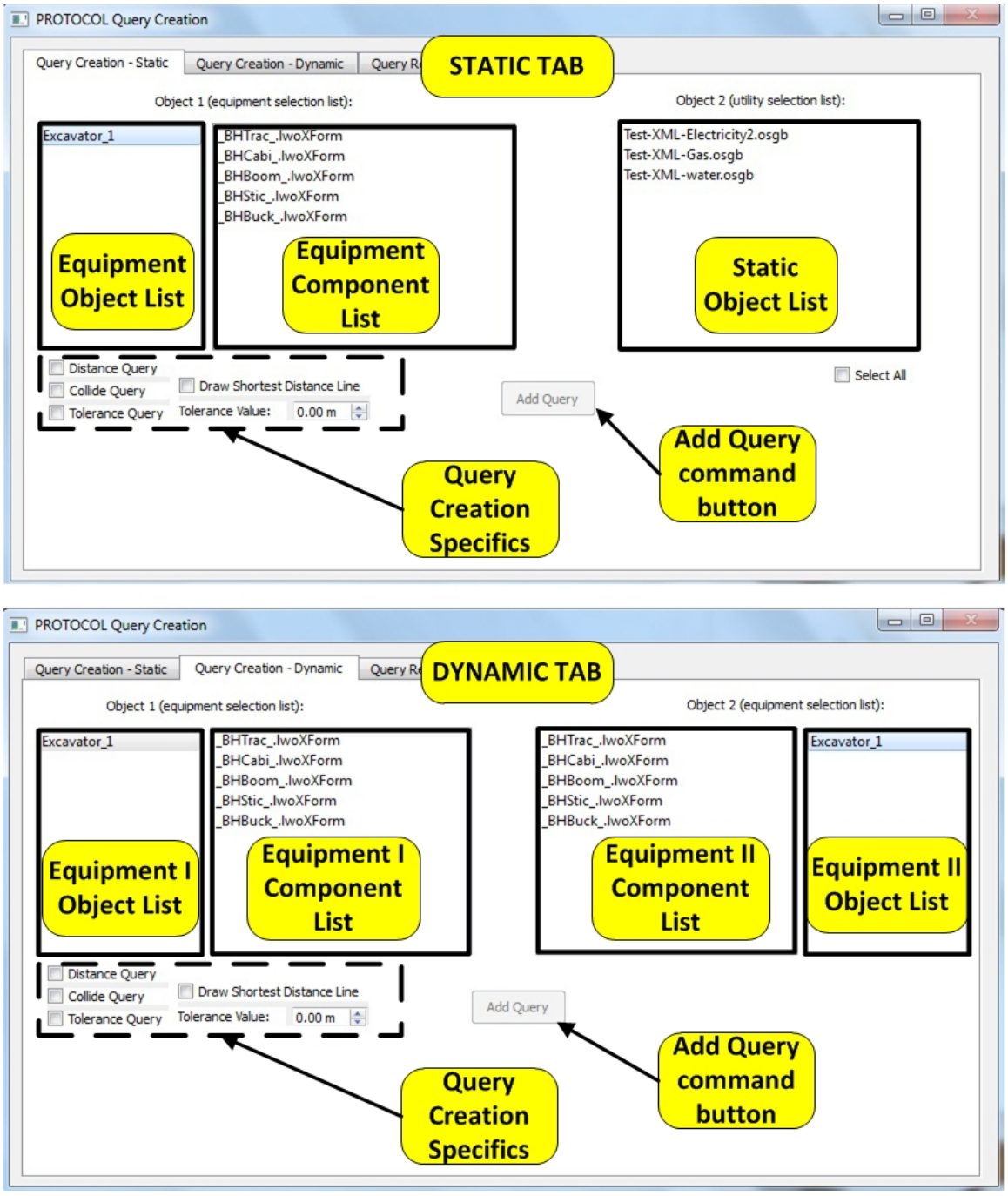

Fig. B.23: PROTOCOL interface lists and commands annotated for static and dynamic tab widgets

Thus users would select a single equipment component from the Equipment Component List, one or more items from the Static Object List, and then use the checkboxes shown in Figure B.23, highlighted by the Query Creation Specifics label. The checkboxes allow users to select the types of queries to include in the PROTOCOL query they are setting up. ‘Distance Query’ monitors the Euclidean distance between the two entities; ‘Collide Query’ informs the user when a pair of entities intersect, and ‘Tolerance Query’ allows users to set up a safety threshold or buffer distance between two entities, and provides warnings when that distance is breached by one of the entities. The final checkbox allows users to select if they require the shortest distance between two entities to be represented by a line in the SeePlusPlus visualization.

After the required dynamic and static entities are selected, and PROTOCOL query details are checked, the query can be generated using the ‘Add Query’ button, which is highlighted in Figure B.23. As an example, Figure B.24 shows PROTOCOL queries being created between the bucket component of Excavator_1 and all three buried utilities loaded in the HVS scene. The PROTOCOL queries are chosen to include distance, collision, and tolerance computations. A distance of 1.5 meters is set as the tolerance threshold. The option to draw the line joining the closest pair of points between two entities is also checked in the interface.

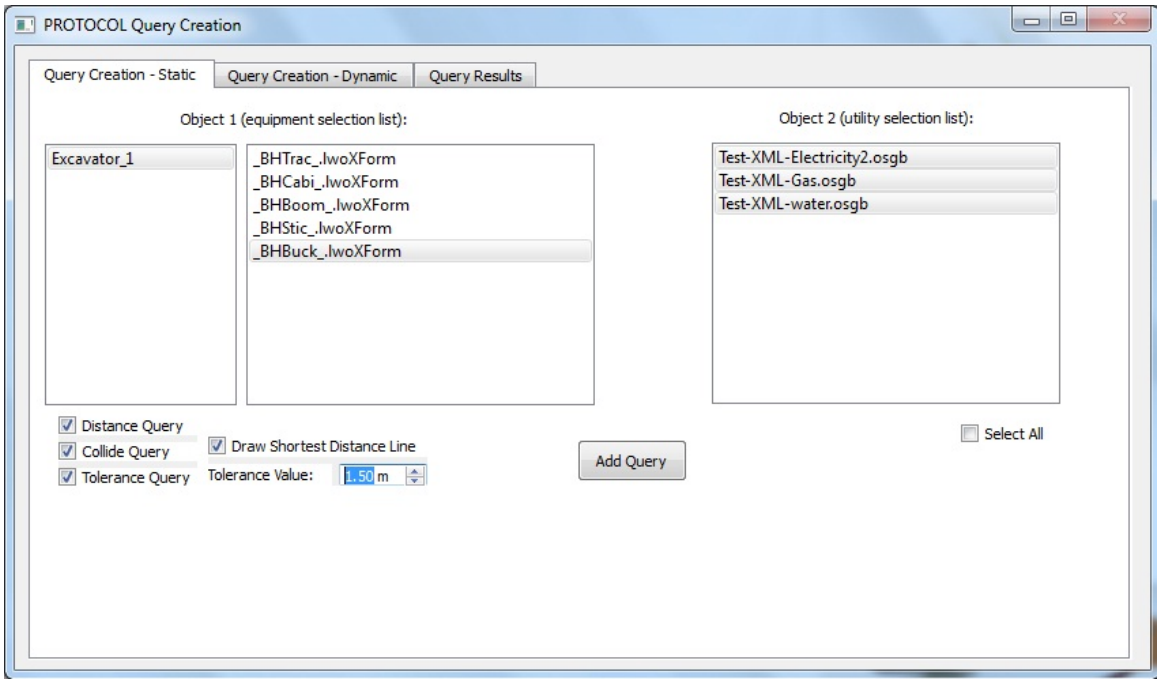

Fig. B.24: PROTOCOL interface showing queries between bucket (dynamic) component and buried utilities (static) for distance, collision, and tolerance queries

Execution of the ‘Add Query’ command through its dedicated button results in a success message displayed to the user, as shown in Figure B.25, if the queries are able to be created by the module. It is important to point out that a common cause for failure of PROTOCOL query creation is the lack of valid ‘.tris’ files that are required for successful PROTOCOL query creation. The ‘.tris’ files contain the number of polygons, as well as a list of all polygon vertices, that form part of every 3D model’s surface. The name of the ‘.tris’ file is derived from the name of the 3D model component, and a mis-named ‘.tris’ file can also lead to a failed PROTOCOL query creation. The failure messages displayed to the user clearly describe which 3D model is the cause of the failed query creation, as shown in Figure B.26.

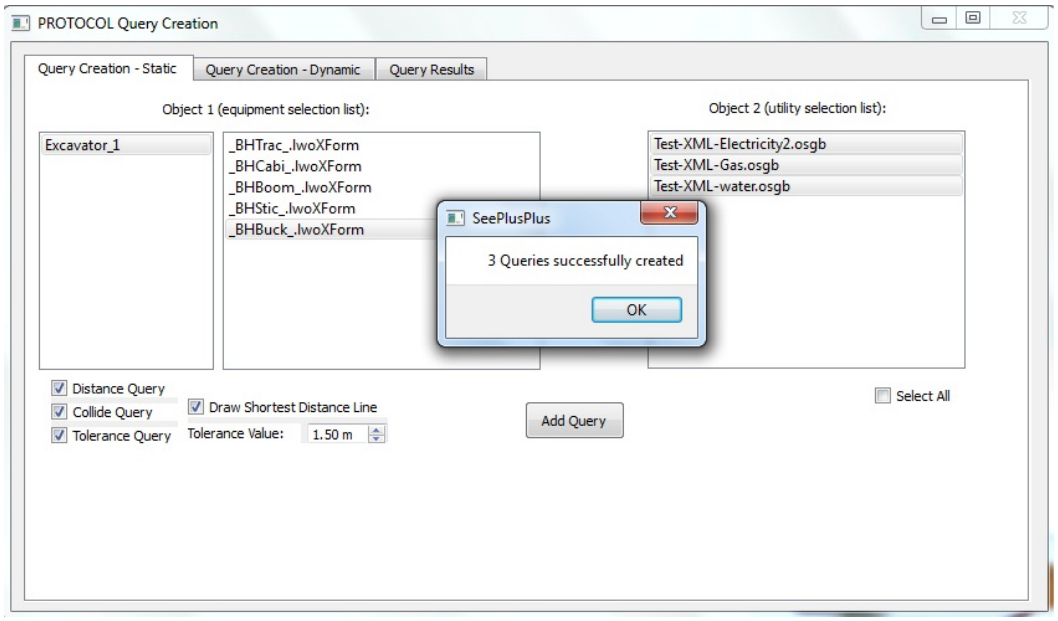

Fig. B.25: PROTOCOL interface displaying success message to user after creation of the three queries

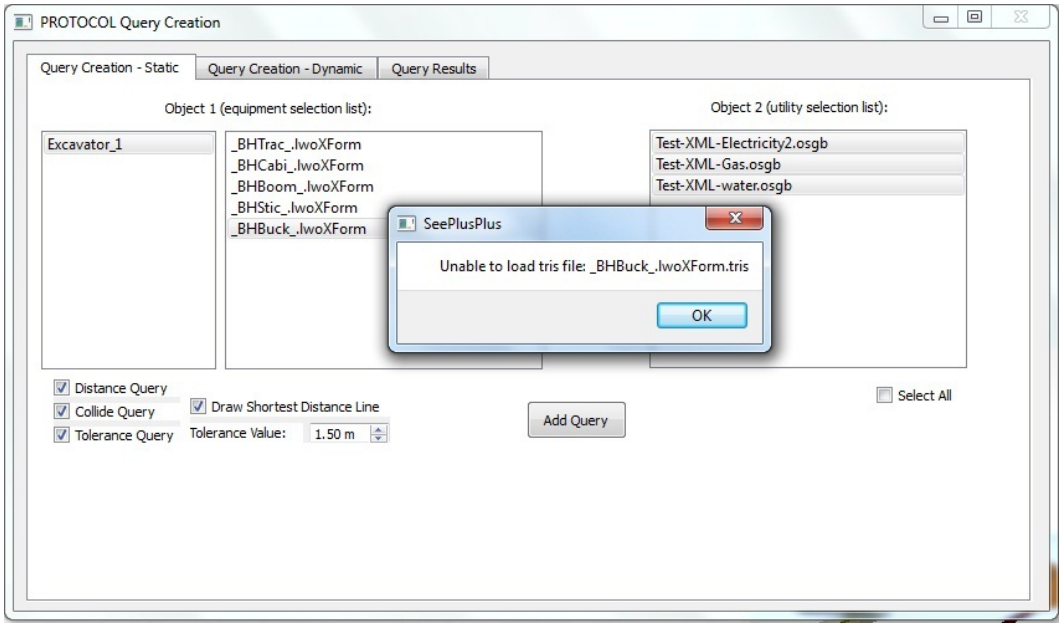

Fig. B.26: PROTOCOL error message displayed to user due to bucket component '.tris' file _BHBuck_.lwoXForm.tris not being found

After PROTOCOL queries are successfully created, the SeePlusPlus interfaces updates its Proximity Distance Display widget at its bottom right from its default/un-initialized value of ‘9999.99’ to the value corresponding to the closest distance between the bucket (dynamic entity) and any of the buried utilities (the static entities). This update is shown in Figure B.27. It must be pointed out that this feature of updating the Proximity Distance Display widget only occurs if the ‘Distance Query’ option is checked at the time of query creation. In addition, the ‘Draw Shortest Distance Line’ option was also checked at the time of PROTOCOL query creation. Hence Figure B.28 shows the SeePlusPlus interface with shortest distance lines rendered between the bucket tip and each of the surrounding utilities.

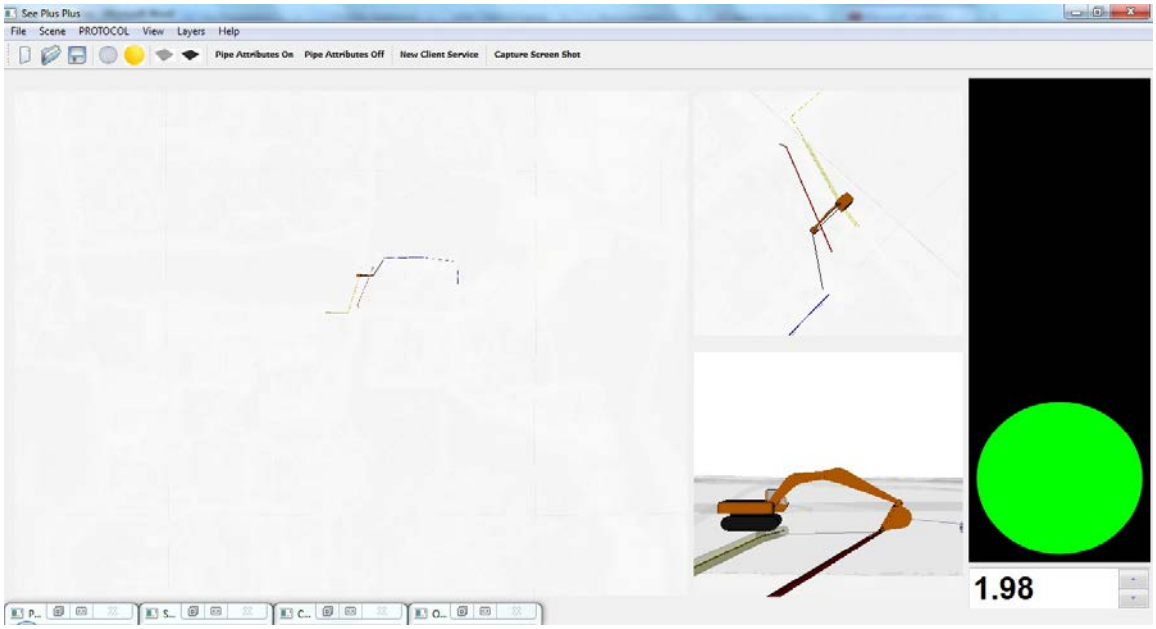

Fig. B.27: SeePlusPlus interface showing Proximity Distance Display with updated value

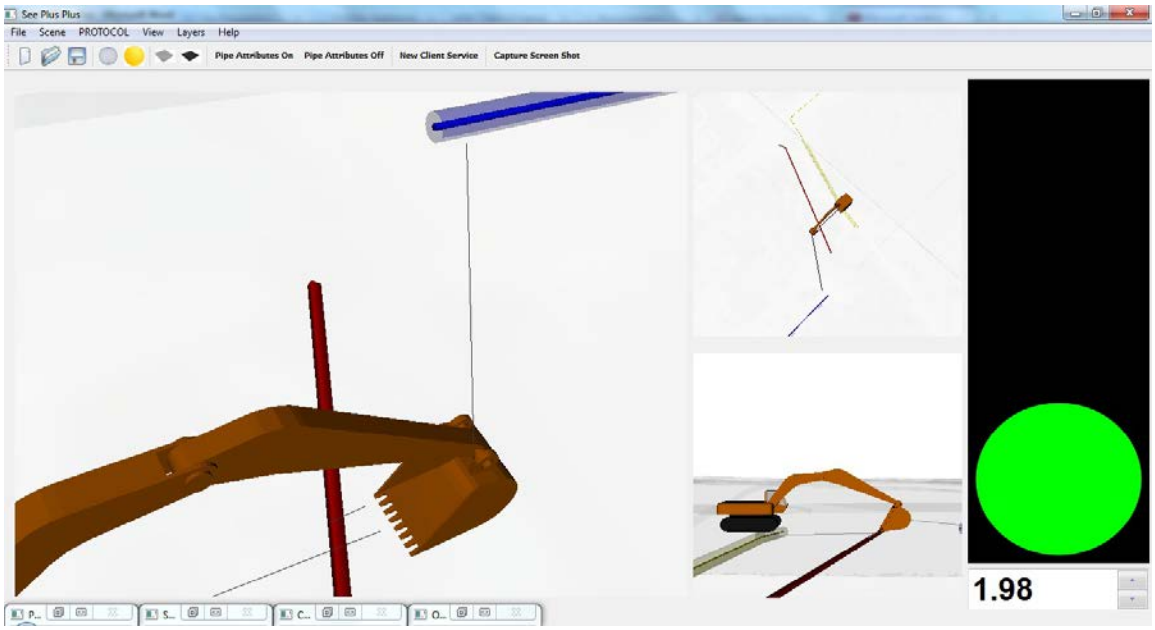

Fig. B.28: SeePlusPlus interface showing lines drawn between bucket component and electric (red), water (blue), and gas (yellow) utilities

The PROTOCOL queries that were created in the steps above can be saved to a user-defined location on the user's system using the 'Save Queries' command under the 'PROTOCOL' menu, which is shown in Figure B.29.

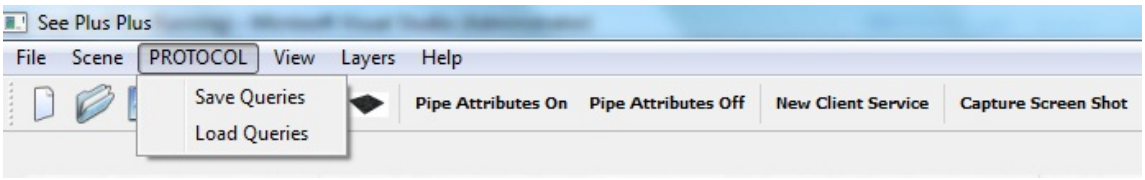

Fig. B.29: PROTOCOL menu

PROTOCOL queries are saved/archived using a PROTOCOL Query File. Such files have a '.pqf' file extension. Execution of the 'Save Queries' command presents the user with a File Save dialog where the filename and location of the .pqf file can be set as shown in

Figure B.30. A user is informed of success through a message, as displayed in Figure B.31.

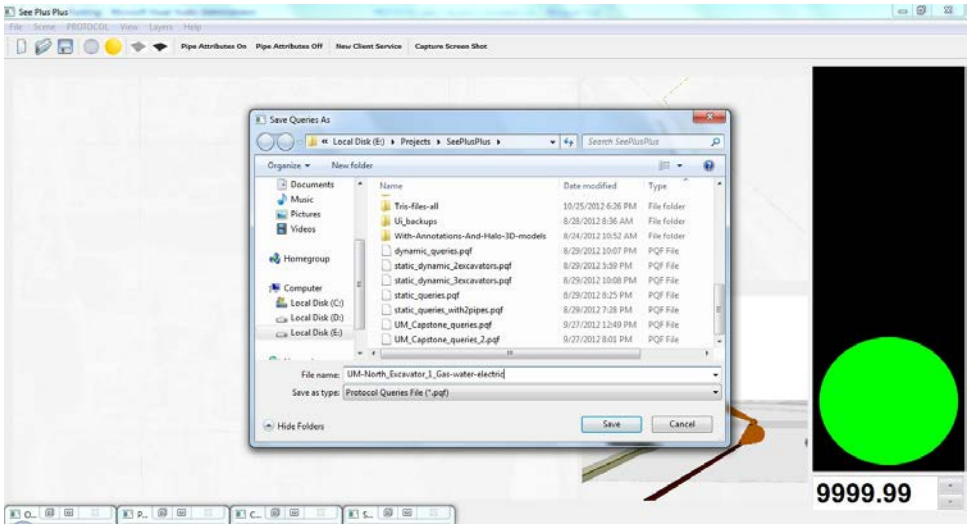

Fig. B.30: File save dialog to allow the user to set the location and file name for the Protocol Query File (.pqf)

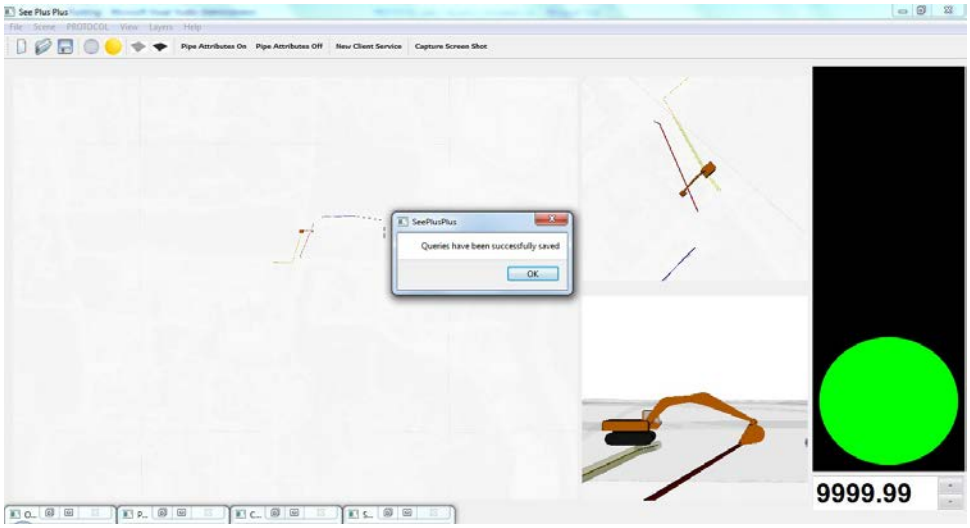

Fig. B.31: Success message displayed to the user after PROTOCOL queries successfully saved to the user's system

PROTOCOL also allows users to restore or reload the saved queries through the ‘Load Queries’ command, as shown in Figure B.29. This command is designed so that an HVS scene and its associated queries can be pre-created and deployed at run-time in the field with minimum effort and by personnel not trained in advanced computer graphics. Execution of the ‘Load Queries’ command presents users with a File Open dialog box through which the ‘.pqf’ file corresponding to the current operation and jobsite can be loaded, as shown in Figure B.32. When all of the queries in the ‘.pqf’ file are successfully loaded, the user is informed of the command’s success, as shown in Figure B.33.

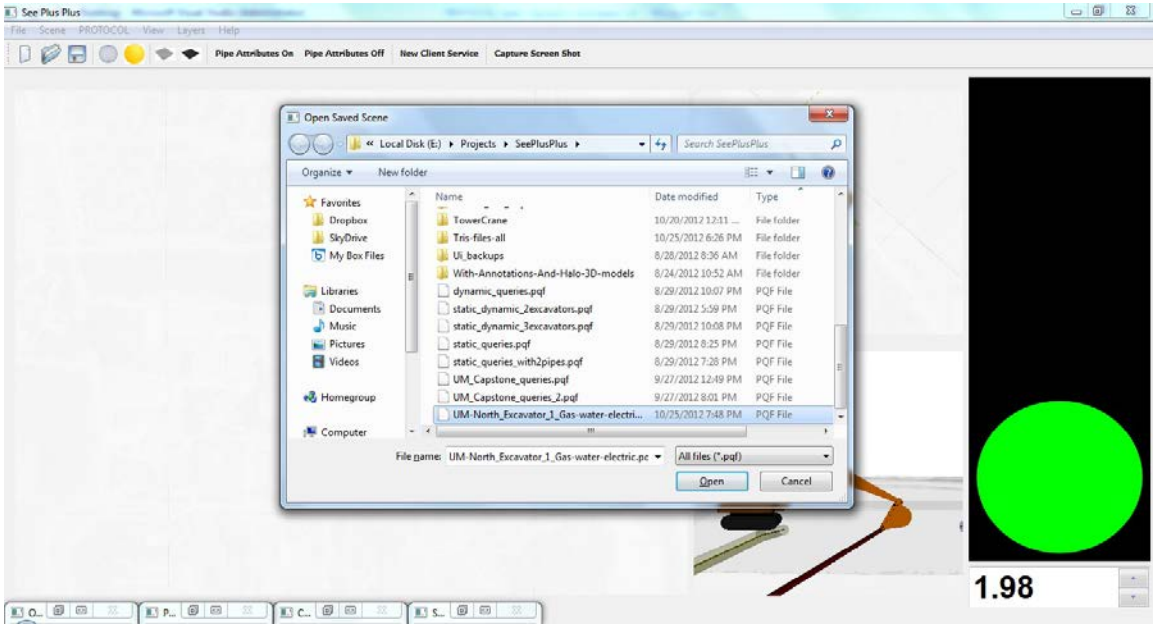

Fig. B.32: File open dialog to allow user to select Protocol Query File (.pqf) corresponding to the user’s saved queries

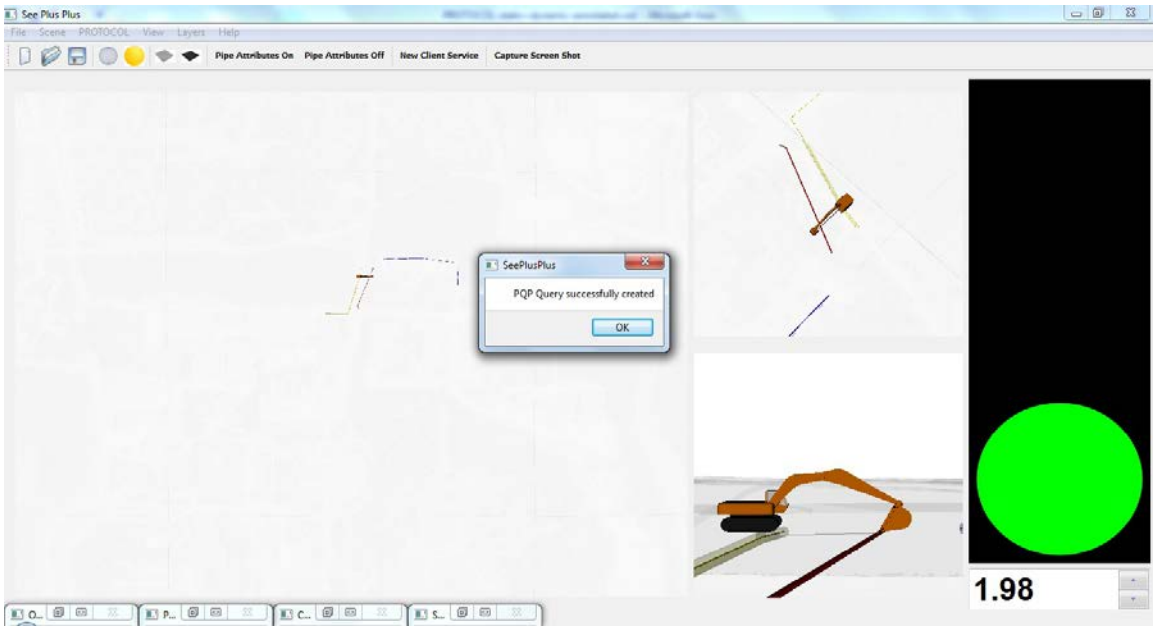

Fig. B.33: PROTOCOL module displaying success message after saved queries successfully applied to the HVS scene

B.4 Setting up server-client connections in Sensor Stream Acquisition Allocation (S2A2) Framework

The SeePlusPlus application is able to represent the dynamic entities in a real-world construction operation inside of a 3D virtual environment through the S2A2 framework. The S2A2 framework presents the user with an interface to map individual sensor data streams from the real world to specific equipment components in the virtual world. S2A2 is implemented as a graphical interface and is present as a module inside SeePlusPlus, as seen in Figure B.34. The HVS scene loaded in SeePlusPlus is similar to the one used thus far in Appendix B. It consists of an excavator (Excavator_1), three buried utilities (electric, water, and gas), and a 3D terrain surface. The S2A2 interface shows the list of 3D equipment objects and the available components of each equipment object. Figure B.35 shows a close-up view of the S2A2 interface.

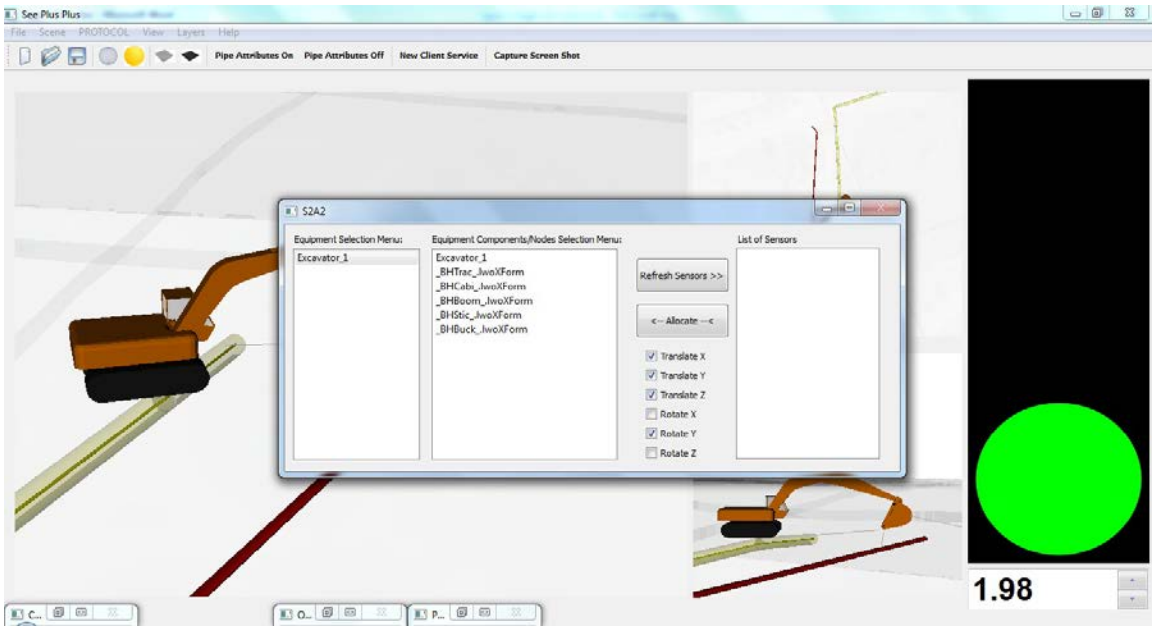

Fig. B.34: S2A2 module interface present inside SeePlusPlus application

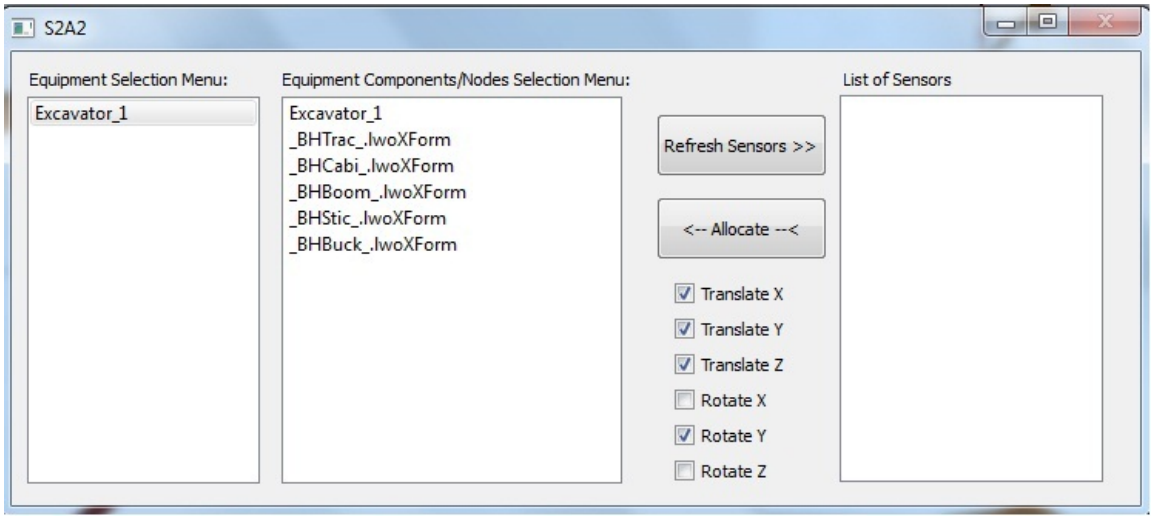

Fig. B.35: S2A2 interface showing excavator object and its associated components

Figure B.36 shows a detailed view of the S2A2 interface with individual functions highlighted and interface components labeled. The S2A2 interface essentially consists of two halves or sides—the equipment half and the sensor half. The equipment half itself

consists of two lists—the left-most list and the right-most list. The left-most list shows the number equipment objects present in the scene. In the case of the HVS being demonstrated, there exists only Excavator_1 in the scene. The right-most list (i.e., Equipment Component List) exposes the number of individual components making up the equipment object. Thus in the case of Excavator_1, the components that constitute it are track, cabin, boom, stick, and bucket (i.e., the file names _BHTrac_.lwoXForm, _BHCabi_.lwoXForm, _BHBoom_.lwoXForm, _BHStic_.lwoXForm, and _BHBuck_.lwoXForm).

The sensor half, which is the right-most list, displays the number of sensor servers that are connected to the SeePlusPlus application. Figure B.36 shows the Sensor List being empty, as no sensor server applications are connected to the client application (SeePlusPlus) in this case. Details on connecting a sensor server application are provided in the following paragraphs. The part of the S2A2 interface between the equipment half and the sensor half consists of the ‘Allocate’ command button, which upon execution connects (or maps) a single sensor data stream to a single equipment component. The mapping is one-one and influenced by the user through the Position Orientation update selection checkboxes, as shown in Figure B.36. There are six potential updates that can be given to any equipment component. The first is translation along the X-, Y-, and/or Z-axis. The second is rotation about the X-, Y-, and Z-axis (i.e., roll, pitch, and yaw, respectively). When a user selects one of these six options, that particular component of the sensor data stream is used to update the equipment component’s position and/or orientation. For example, as seen in Figure B.36, the options checked are translation

about X-, Y-, and Z-axis along with rotation about the Y-axis. Thus an equipment component would experience a translation update along the three axes as well as a rotation about the Y-axis (pitch rotation) as a result of any sensor stream that would be present in the Sensor List. The 'Refresh Sensors' command button updates the Sensor List in the interface and removes any inactive sensor streams, making available only those sensor streams that are currently transmitting data from the real world.

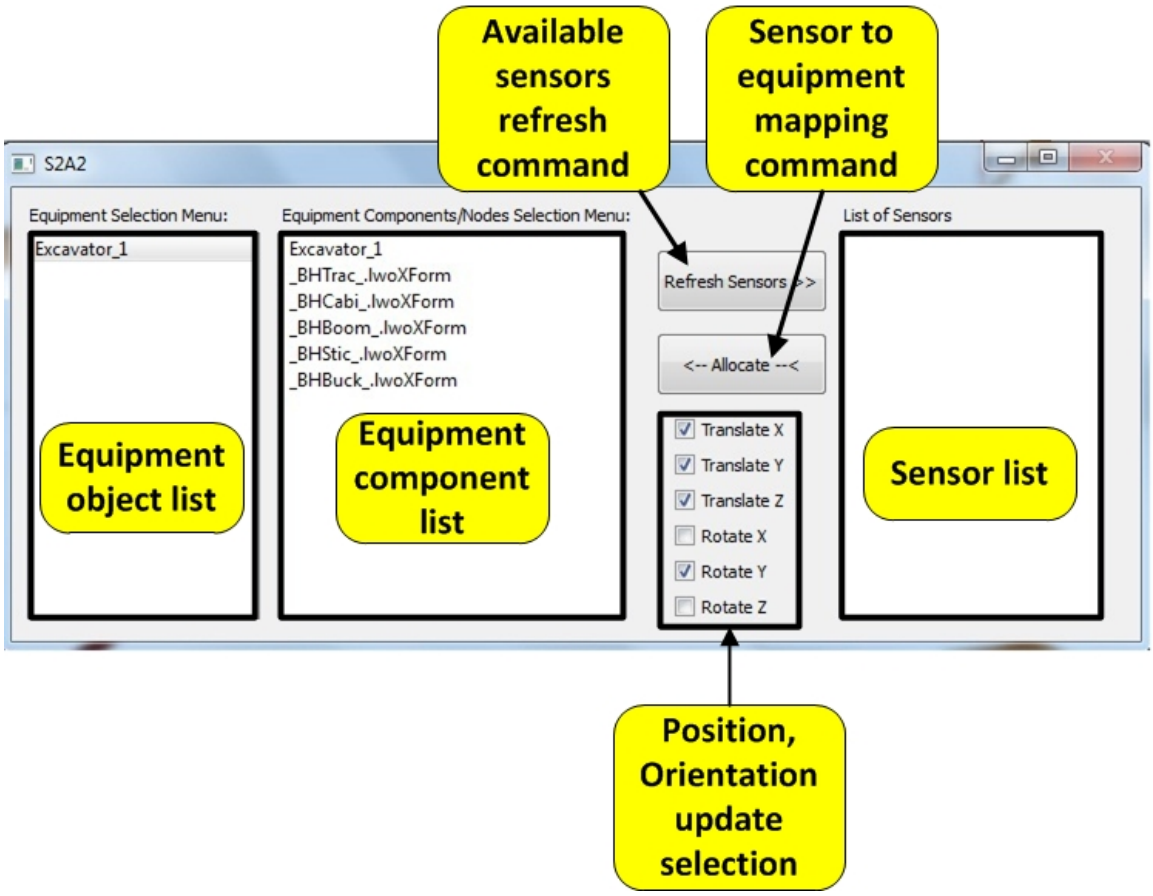

Fig. B.36: S2A2 interface lists and commands annotated for reference

The previous sections have described the details of the S2A2 interface and the commands for mapping sensor data streams to equipment components. However, as the user, it is

essential to know how a sensor data stream can be made available to the S2A2 interface. The connection between real-world sensor data and its availability in SeePlusPlus and the S2A2 interface consists of two parts: 1) the sensor server application, and 2) the client application. The sensor server is any application that converts sensor data into a data format that can be used to update 3D equipment components in SeePlusPlus. This data structure used to transmit sensor data from the server to client application is a struct (programming language data structure) called PositionOrientationStruct. As its name suggests, it is a container for storing the 3 aspects of position (x, y, z) and orientation (roll, pitch, yaw). This convention is commonly used to describe position and orientation, and is thus adopted. The struct consists of six floating point values that represent the six position and orientation values. The makeup of the PositionOrientationStruct is shown in the code sample below—

```
//this struct is used to transfer position(x, y, z) data
//and orientation(x, y, z) from server to client
struct PositionOrientationStruct
{
  float xCoordinate;
  float yCoordinate;
  float zCoordinate;
  float roll;
  float pitch;
  float yaw;
};
```

The server application transmits this struct through a socket-based interface to the client application. The client-side application in turn connects with the server-side and makes the sensor data stream available in the S2A2 interface for allocation to 3D equipment

components. The command to initiate a client-side connection, termed as a client service, is present in the SeePlusPlus interface, as highlighted in Figure B.37.

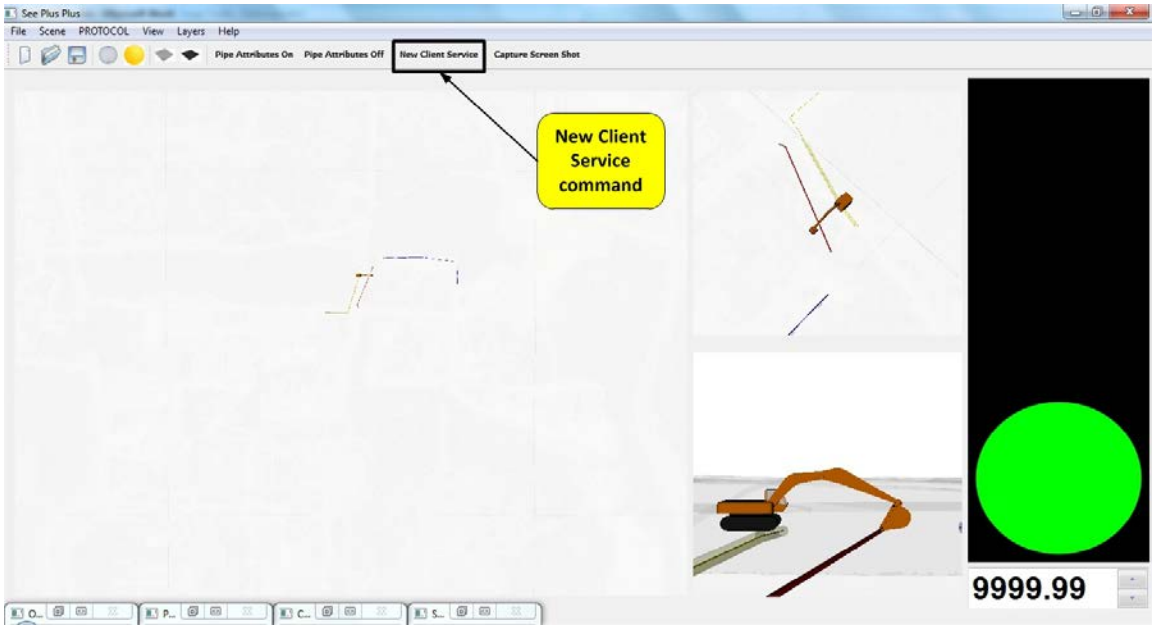

Fig. B.37: New Client Service command button highlighted in SeePlusPlus interface

Before the user initiates a connection to a server-side application, the application on the server-side must be instantiated and made ready to accept connection requests from the client-side. The code sample below shows an example of a server-type application to transmit a PositionOrientationStruct data object to a client application. On a Windows operating system, the winsock library ‘ws_32.lib’ needs to be linked to any application based on socket connections. The executable obtained after the following code is compiled successfully would be used to complete the rest of the steps that describe the procedure for completing the server-client connection.

```
#include <stdio.h>
#include <winsock2.h>
#include <windows.h>
#include <ctime>
#include <iostream>
#include <fstream>


//this struct is used to transfer position(x, y, z) data
//and orientation(x, y, z) from server to client
struct PositionOrientationStruct
{
  float xCoordinate;
  float yCoordinate;
  float zCoordinate;
  float roll;
  float pitch;
  float yaw;
};


int main(int argc, char* argv[])
{
  if(argc < 2)
  {
    std::cout << "Usage: TCP_Server_application_2.exe <port-number>
<frequency-seconds>\n"
              << "where <frequency-seconds> milliseconds to wait
between successive sends\n";
    exit(0);
  }

  //output file stream
  std::ofstream outStream;
  outStream.open("server_stream.txt");

  //Initialize data structure to send over socket connection
  PositionOrientationStruct* positionOrientationStructPointer = new
PositionOrientationStruct();
  positionOrientationStructPointer->xCoordinate = 0.0f;
  positionOrientationStructPointer->yCoordinate = 0.0f;
  positionOrientationStructPointer->zCoordinate = 0.0f;
  positionOrientationStructPointer->roll = 0.0;
  positionOrientationStructPointer->pitch = 30.0;
  positionOrientationStructPointer->yaw = 45.0;

  //get the sensor type from user
  unsigned int userEnteredSensorType = strtoul(argv[2], 0, 0);


  // Initialize Winsock.
  WSADATA wsaData;
  int iResult = WSAStartup(MAKEWORD(2, 2), &wsaData);
  if (iResult != NO_ERROR)
  {
    printf("Server: Error at WSAStartup().\n");
    return -1;
```

```
  }
  else
  {
    printf("Server: WSAStartup() is OK.\n");
  }


  //Create a SOCKET for listening for incoming connection requests.
  //The socket in the listenin state is used in the accept()
  SOCKET listenSocket;
  listenSocket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);

  if (listenSocket == INVALID_SOCKET)
  {
    printf("Server: Error at socket(): %ld\n", WSAGetLastError());
    WSACleanup();
    return -1;
  }
  else
  {
    printf("Server: socket() is OK.\n");
  }


  //Initialize sockaddr_in struct
  //The sockaddr_in structure specifies the address family,
  //IP address, and port for the socket that is being bound.
  unsigned short userEnteredPortNumber = strtoul(argv[1], 0, 0);
  sockaddr_in service;
  service.sin_family = AF_INET;
  service.sin_addr.s_addr = inet_addr("127.0.0.1");
  service.sin_port = htons(userEnteredPortNumber);


  //Bind the socket
  if  (bind(listenSocket,  (SOCKADDR*)  &service,  sizeof(service))  ==
SOCKET_ERROR)
  {
    //printf("Server: bind() failed.\n");
    printf("bind failed with error: %ld\n", WSAGetLastError());
    closesocket(listenSocket);
    WSACleanup();
    return -1;
  }
  else
  {
    printf("Server: bind() is OK.\n");
  }

  // Listen for incoming client connection requests on the created
socket
  if (listen(listenSocket, 10) == SOCKET_ERROR)
  {
    printf("Server:    listen()    failed    with    error:    %ld\n",
WSAGetLastError());
    closesocket(listenSocket);
    WSACleanup();
```

```
    return -1;
  }
  else
  //returns zero if no error occurs
  {
    printf("Server: listen() is OK.\n");
  }


  // Create a SOCKET for accepting incoming requests.
  //The result from accept() will be stored in this socket
  SOCKET acceptSocket;
  //initialize socket to INVALID_SOCKET value
  acceptSocket = INVALID_SOCKET;
  printf("Server: Waiting for client to connect...\n");

  // Accept the connection if any...
  //The program will freeze/wait at accept() until a client program is
accepted by server
  //(Use asynchronous sockets to avoid this)
  while(acceptSocket == INVALID_SOCKET)
  {
    acceptSocket = accept(listenSocket, NULL, NULL);
  }
  printf("Server: accept() is OK.\n");
  printf("Server: Client connected...ready for communication.\n");

  //The server socket listenSocket is no longer needed so closing it
  //If you require >1 clients to connect then you would keep this
socket open
  //and create additional accept sockets using this listenSocket
  closesocket(listenSocket);


  //SEND code
  //Send data to client application using send()
  int secondsToWait = strtoul(argv[3], 0, 0);
  FILETIME nowTime;
  time_t secondsS2A2Handler;
  time_t milliSecondsFromFileTime;
  ULARGE_INTEGER ull;
  int bytesSent = -1;

  unsigned int count = 0;
  bool ascend = true;
  while(true)
  {
    bytesSent                    =                    send(acceptSocket,
(char*)positionOrientationStructPointer,
                    sizeof(PositionOrientationStruct), 0);

    if(ascend)
    {
      if(positionOrientationStructPointer->pitch >= 90.0)
      //Boom = 60, Stick = 30, Buck = 15
      {
        ascend = false;
```

```
      }
      else
      {
       positionOrientationStructPointer->pitch += 0.5;
      }
    }
    if(!ascend)
    {
      if(positionOrientationStructPointer->pitch <= -90.0)
      //Boom = 20, Stick = -30, Buck = -15
      {
        ascend = true;
      }
      else
      {
        positionOrientationStructPointer->pitch -= 0.5;
      }
    }

    Sleep(secondsToWait);
    count++;
  }

  outStream.close();
  WSACleanup();
  return 0;
}
```

This server application requires the user to specify the port number and time delay between successive transmission cycles (expressed in milliseconds). As shown in the code sample below, the server-side application is executed with a port number of 5150 and 100 millisecond time delay between successive transmission cycles. The server-side application is initialized and ready to accept a connection request from the client side.

```
E:\Projects\TCP_Server_Application_2\Debug>TCP_Server_Application_2.exe
5150 100

Server: WSAStartup() is OK.
Server: socket() is OK.
Server: bind() is OK.
Server: listen() is OK.
Server: Waiting for client to connect...
```

The connection from the client-side is started by executing the ‘New Client Service’ command button. This presents the user with an interface, as shown in Figure B.38. The interface has three fields for user input: 1) field one—the IP address of the server application machine (in the case of this example, both server and client applications are running on the same machine and hence the value ‘127.0.0.1’; 2) field two—the port number through which the client application requests the connection (this would be 5150 as used while initializing the server-side application); and 3) field three—an identifier for the client service sensor stream as it would appear in the S2A2 interface Sensor List. On confirming the user input, a series of success or failure messages are presented to the user to inform them of a successful connection or of specific errors during the server-client connection process.

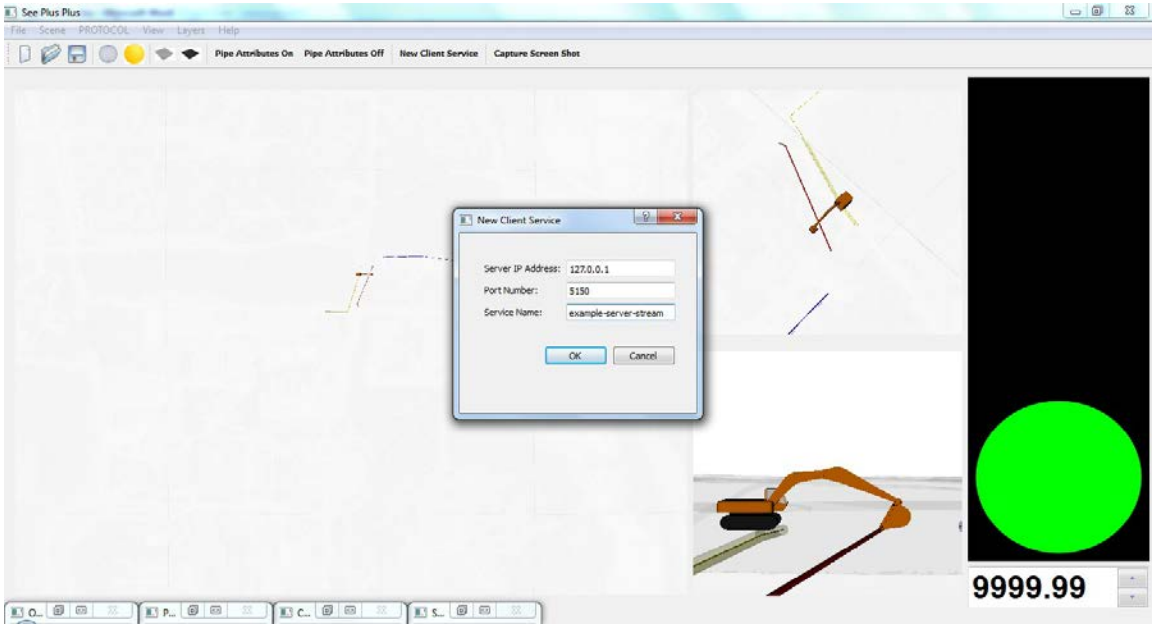

Fig. B.38: New Client Service interface for getting user input

After the server-client connection is successfully established, the newly connected sensor server will now appear in the S2A2 interface Sensor List, as shown in Figures B.39 and B.40.

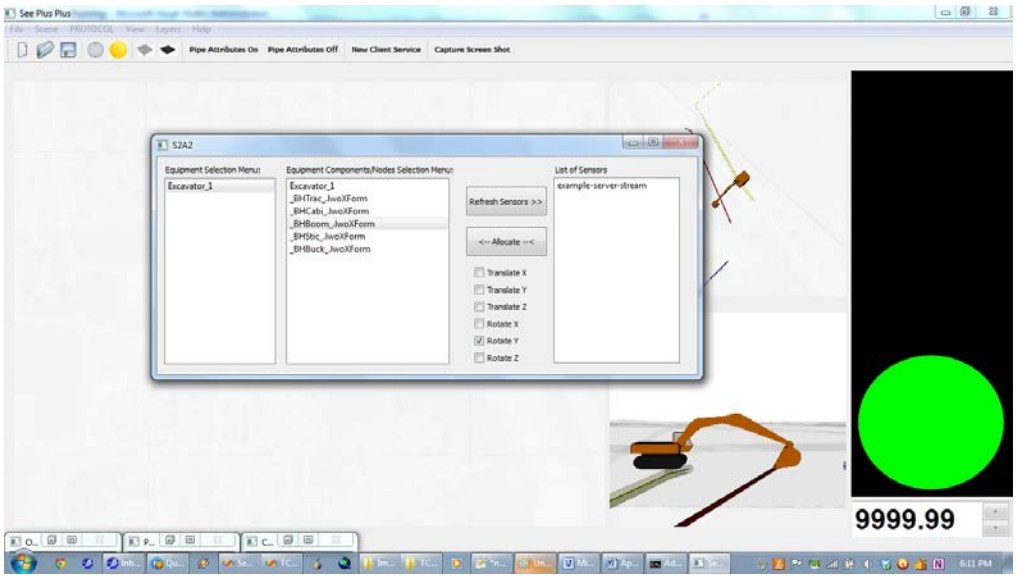

Fig. B.39: S2A2 interface showing updated Sensor List within SeePlusPlus after successful server-client connection

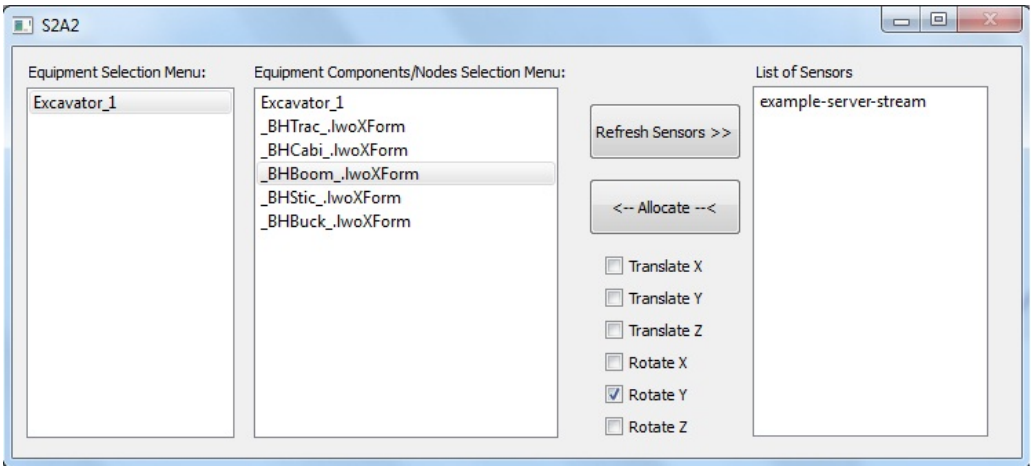

Fig. B.40: S2A2 interface detailed view showing 'example-server-stream' in the updated Sensor List

Any item in the Sensor List corresponds to an active sensor data stream that can be mapped to any equipment component in the Equipment Component List. This is a one-one mapping and is accomplished after 1) a single element from the Equipment Component List and a single element from the Sensor List are selected; and 2) the details of the mapping are completed by selecting one or more options from the Rotate and Translate checkboxes. The server application used in this example provides an update of rotation about the Y-axis. Thus the ‘example-server-stream’ is allocated to Excavator_1’s boom component by executing the ‘Allocate’ command button. After this allocation, the boom component of Excavator_1 will show rotation about the Y-axis corresponding to the server-side update in values. In this way, sensor data streams from the real world can be mapped to 3D equipment components in an HVS using the S2A2 interface in SeePlusPlus.

B.5 Toggling Layers in SeePlusPlus

The SeePlusPlus application is designed such that a combination of static and dynamic entities can be represented in the presence of a 3D terrain model. The terrain, static entities, and dynamic entities can each be viewed as a layer. However, it can be expected that certain scenarios would require one or more of the three layers to be turned off in order to facilitate clearer viewing. The ‘Layers’ menu in SeePlusPlus presents users with toggle options, as shown in Figure B.41, to turn individual layers on/off. Figure B.42 shows screenshots of the same HVS scene in SeePlusPlus with each layer alternately turned on/off.

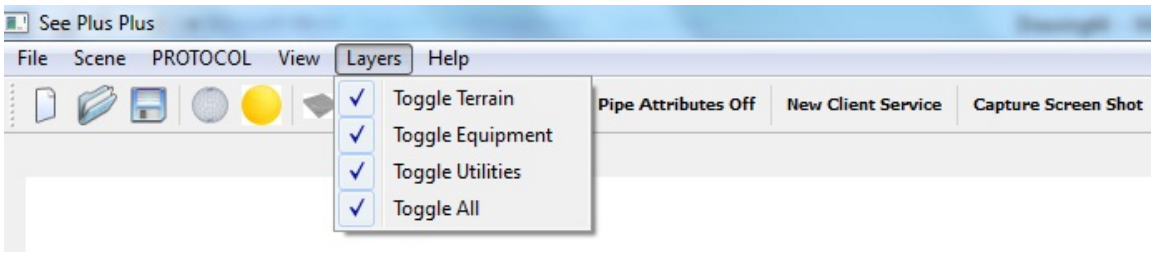

Fig. B.41: The drop-down 'Layer' menu in SeePlusPlus

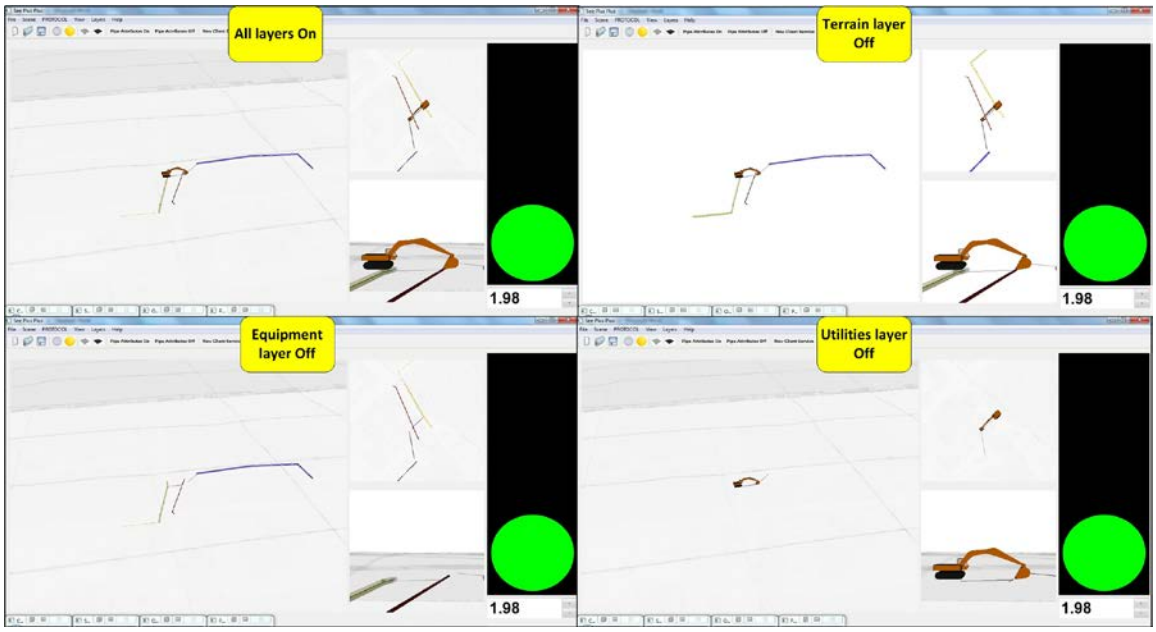

Fig. B.42: Toggling each layer On/Off to show the interactive abilities of SeePlusPlus

B.6 Using the multi-modal output

The warnings from proximity monitoring are presented to the user through a traffic light warning mechanism using a green color to signify safe working, amber to indicate a breach of the tolerance zone, and red to indicate that a collision has occurred. Figure B.43 shows the traffic light widget changing the displayed color from green to amber, and

from amber to red. This color change occurs as the bucket comes closer to the electric (red) utility line.

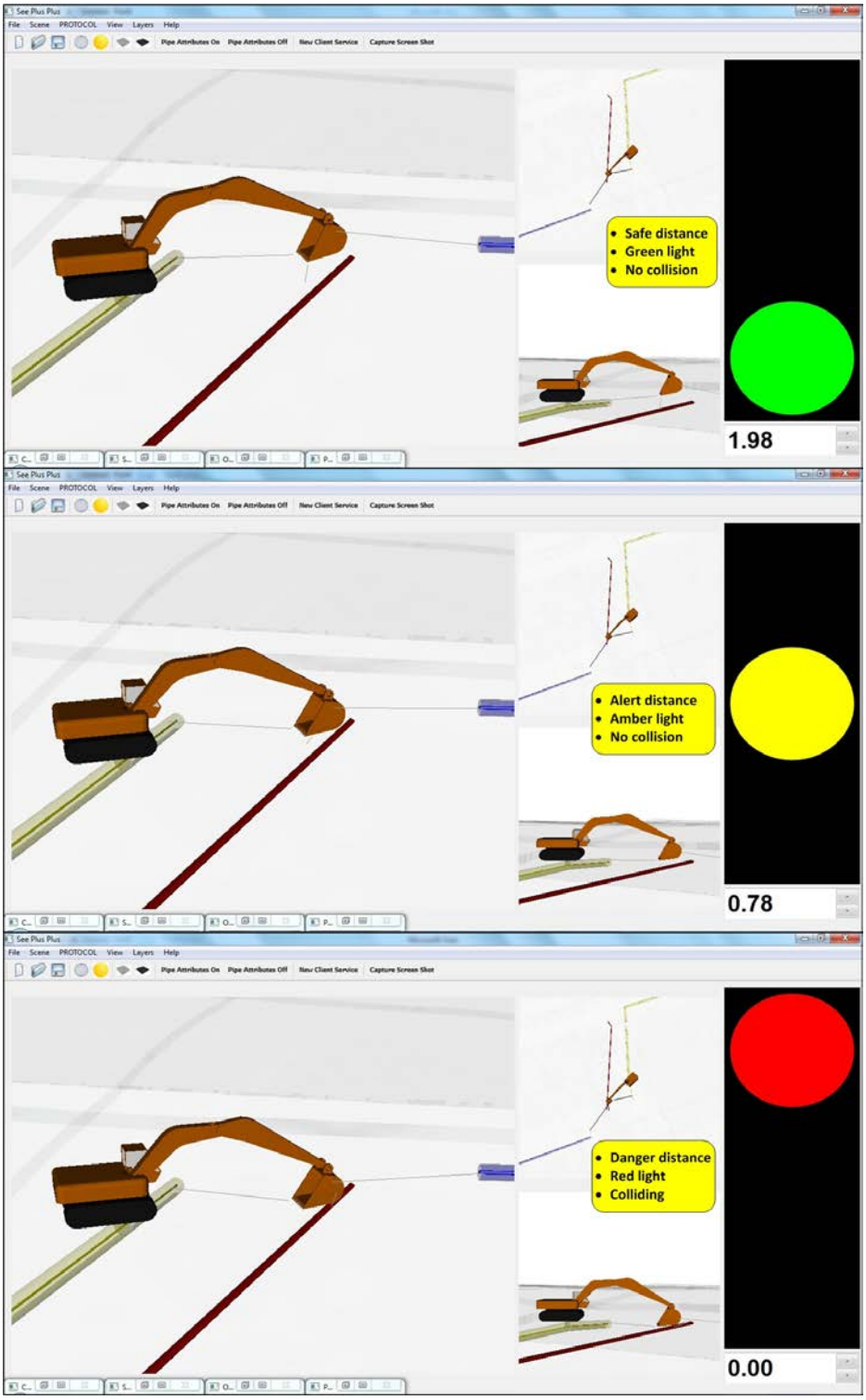

Fig. B.43: Traffic light widget showing green, amber, and red signals

In addition to the traffic light warning mechanism, an audio warning is also provided to the user when the safety threshold is breached to warn against an impending collision or accident. The Proximity Distance Display is designed to display the nearest neighboring entity and thus signifies the closest distance to a potential collision and accident. In this manner, SeePlusPlus not only offers real-time 3D visualization support to the user but also provides audio and visual warning mechanisms as demanded by situations, along with distance information if required by the user.

B.7 Using Object Controller Widget for moving equipment objects

The SeePlusPlus application also provides a method for users to manually control 3D virtual equipment that is loaded in an HVS scene, as shown in Figure B.44. It is expected that such functionality would enable users and scene creators to test the translation and rotation affordances of the components of any equipment object loaded in SeePlusPlus. It can also be used to examine and compare the expected and actual axis of rotation so that sensor data streams and server-side sensor applications can be modified to reflect the actual axis of rotation of the various equipment components.

The Object Controller Widget interface has two lists: 1) on the left is the list of all equipment objects present in the scene, and 2) on the right is the list of all components making up the equipment object selected in the former list. The rotation and translation motions are made possible through push buttons. At the bottom of the interface are fields that display the current rotation angles of the selected equipment component from List 2

(the component list). A detailed view along with functional explanations of the various buttons on the Object Controller Widget is shown in Figure B.45.

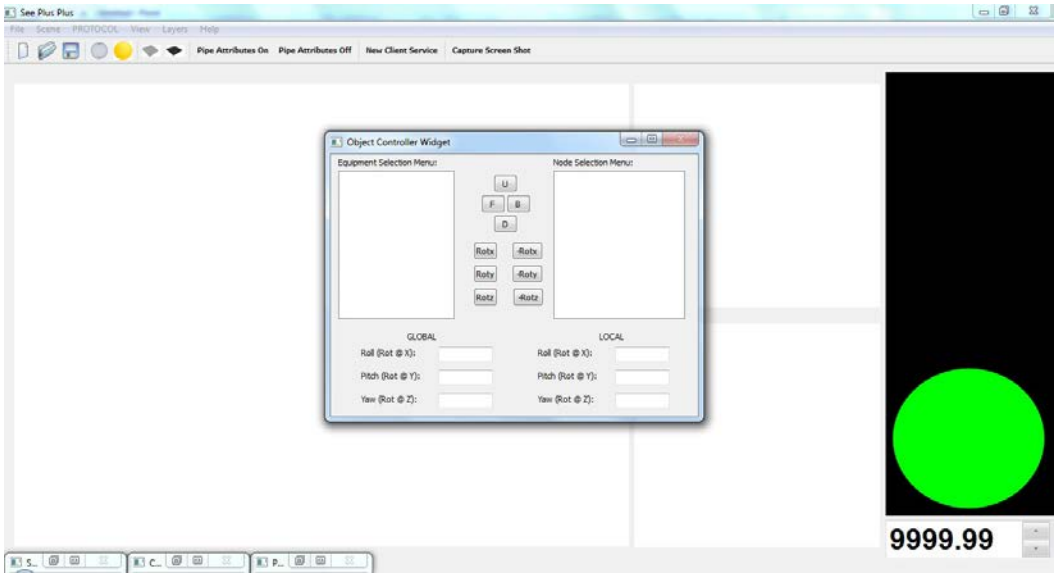

Fig. B.44: Object Controller Widget interface in the SeePlusPlus application

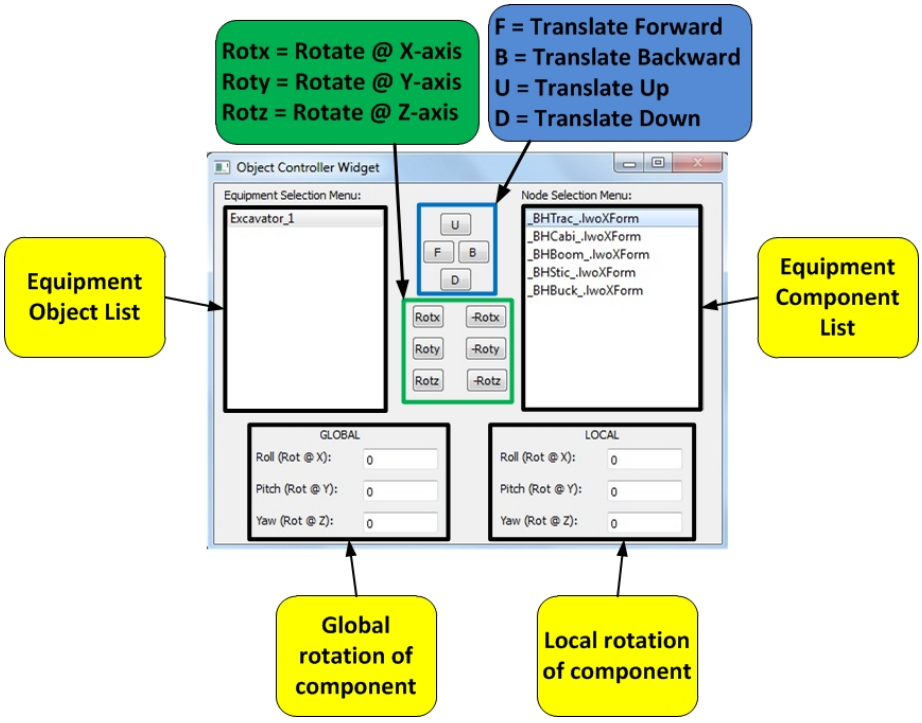

Fig. B.45: Annotated and detailed view of Object Controller Widget