Int J Adv Manuf Technol
DOI 10.1007/s00170-005-0370-9

ORIGINAL ARTICLE

Liang Zhu · Jacob Barhak · Vijay Srivatsan · Reuven Katz

Efficient registration for precision inspection of free-form surfaces

Received: 15 March 2005 / Accepted: 15 November 2005
© Springer-Verlag London Limited 2005

Abstract Precision inspection of free-form surface is difficult with current industry practices that rely on accurate fixtures. Alternatively, the measurements can be aligned to the part model using a geometry-based registration method, such as the iterative closest point (ICP) method, to achieve a fast and automatic inspection process. This paper discusses various techniques that accelerate the registration process and improve the efficiency of the ICP method. First, the data structures of approximated nearest nodes and topological neighbor facets are combined to speed up the closest point calculation. The closest point calculation is further improved with the cached facets across iteration steps. The registration efficiency can also be enhanced by incorporating signal-to-noise ratio into the transformation of correspondence sets to reduce or remove the noise of outliers. Last, an acceleration method based on linear or quadratic extrapolation is fine-tuned to provide the fast yet robust iteration process. These techniques have been implemented on a four-axis blade inspection machine where no accurate fixture is required. The tests of measurement simulations and inspection case studies indicated that the presented registration method is accurate and efficient.

Keywords Optical inspection · Registration between point set and CAD model · ICP algorithm · Approximated nearest node · Simulated measurement

This work was supported primarily by the NSF Engineering Research Center for Reconfigurable Manufacturing Systems as part of the Engineering Research Centers Program of the National Science Foundation under NSF Award Number EEC 95-29125.

L. Zhu (✉) · J. Barhak ·
V. Srivatsan · R. Katz
NSF Engineering Research Center for Reconfigurable Manufacturing Systems, College Of Engineering,
University Of Michigan, 2250 G.G. Brown Building,
2350 Hayward Street, Ann Arbor,
MI 48109-2125, USA
e-mail: zhuliang@umich.edu
e-mail: jbarhak@umich.edu
e-mail: vsrivats@umich.edu
e-mail: reuven@umich.edu

Abbreviations BIM: Blade inspection machine · CAD: Computer aided design · CMM: Coordinate measuring machine · ICP: Iterate closest point · MCS: Measurement coordinate system · PCS: Part coordinate system · RMS: Root mean square · SNR: Signal to noise ratio

1 Introduction

Precision inspection is prevalent in manufacturing to verify that the geometric dimensions and tolerances of parts meet the quality requirements. Under ever-increasing demands on improving product quality and reducing production cycle time, inspection is compelled to become fast, accurate, and cost-effective. As of today, coordinate measuring machines (CMM) are still the most widely used equipment in inspection industry. A CMM is essentially a Cartesian robot with one tactile probe. While the CMM measures the parts in high precision, conventional tactile probe is often limited in scanning speed and can not cover features that are smaller than the stylus diameter [1–3]. The tactile probe is especially cumbersome in measuring parts with free-form surfaces as it is difficult to maintain continuous contact with the surfaces. In these cases, non-contact measurements are commonly used for fast acquisition of surface shape without physically touching the part. The actual speed and accuracy of non-contact measurements vary with different technologies. For instance, the x-ray or CT scan can reveal the inside structure blocked by the outside surface, but have relatively coarse accuracy. On the other hand, a variety of modern non-contact range scanners, ranging from pattern projection to laser probes, offer improved inspection speed and accuracy [4].

Achieving speed, accuracy, and repeatability is the major challenge to a portable and flexible inspection system. Ideally, such a system would automatically collect the measurement and make detailed, full-scale comparisons between the measured data and its original design. With this capability installed on machine tool and assembly lines, manufacturing processes can be monitored and

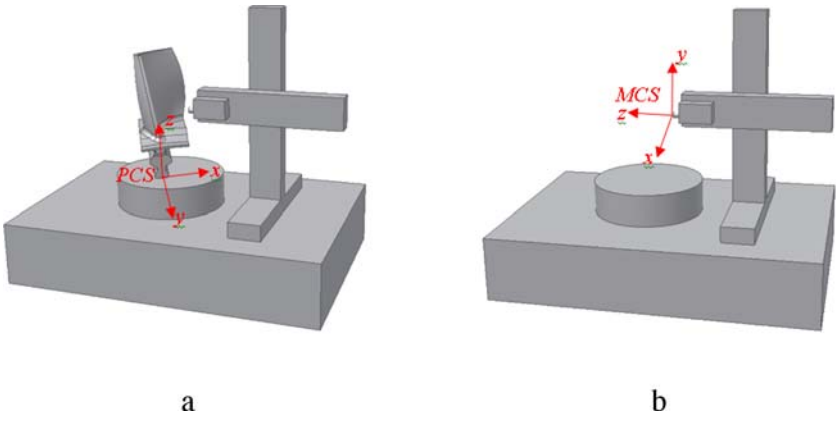


Fig. 1 Different coordinate systems in inspection. **a** Measurement coordinate system. **b** Part coordinate system

controlled in a real-time fashion. In order to achieve automatic and real-time inspection, one of the most challenging problems is to develop an efficient method that aligns the measurement coordinate system and the part coordinate system.

Figure 1 shows the different coordinate systems in inspection. The inspection data is acquired from an optical sensor mounted on the machine. As such, the acquired coordinate value is based on the measurement coordinate system (MCS) as indicated by Fig. 1a. However, the part model is designed in its own coordinate system, named the part coordinate system (PCS), as shown in Fig. 1b. For regular dimensions such as diameter or length, the inspection can be performed based on the direct comparison between the inspected dimensions and the specified values; such are the cases of straight surface tolerances like flatness. However, parts with free-form surfaces do not have a single inspection value. In order to compare the inspected shape with the designed part, it is necessary to find the transformation matrix that will place the inspection data in the appropriate orientation, known as registration in Fig. 2.

In order to register the inspection data, industry practices often adopt a high precision fixture to locate the part orientation. In this approach, the part is closely attached to the fixture. The fixture normally includes some precisely manufactured features (e.g., balls). While the part is being inspected, these fixture features are included as part of the inspection data. Providing the simple geometry of these fixture features, it is relatively easy to find the homogeneous transformation matrix between the machine coordinate system and the part/design coordinate system [5]. The same transformation matrix is also applied to the part data and thus completes the appropriate orientation. While computationally straightforward, the acquired transform matrix from this fixture-based approach is affected by the assembly tolerance between the fixture and the part. In high precision inspection, it is often required to test many iterations of fixture installation to compensate this fixture assembly error, which can be very time-consuming. Plus, a fixture with precise artifacts is expensive or not available at all in some cases.

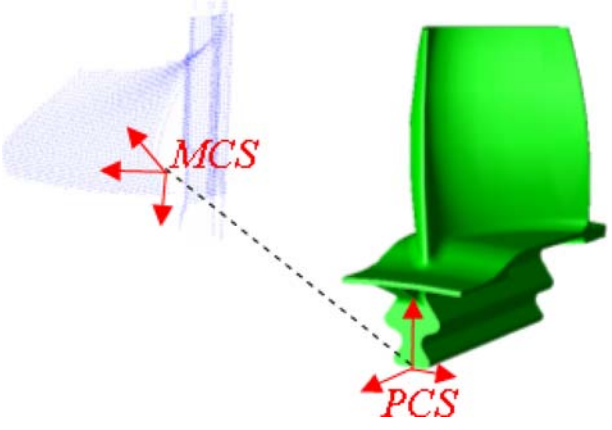


Fig. 2 Inspection registration

Alternatively, the inspection data can be registered to the design model based on the part geometry itself. Geometry-based registration methods are abundant in computer graphics literature. However, many of these approaches are proposed for the purpose of graphics where the accuracy is not critically restricted. It remains unclear how effective these model-based registration methods are in the inspection, especially in the high precision inspection. In this paper, the emphasis of the geometry-based registration will be discussed with respect to their efficiency in precision inspection. Section 2 will review different techniques of available registration methods. Based on these methods and their comparison, a new implementation of geometry-based registration is presented in Section 3. Section 4 will introduce a four-axis optical inspection system incorporated with the implemented registration method. Finally, conclusions are summarized in Section 5.

2 Registration review

The measurement is registered to the part model by matching the correspondence set between the measurement and the part model. The correspondence set can be geometry elements such as holes, edges, and corners. These geometry elements are recognized from the

measurement and matched to the counterpart on the part model [6]. For the parts without distinctive geometry elements such as free-form surfaces, the surface measurements can be grouped into a set of features representing the local geometry characteristic. For instance, Chua and Jarvis [7] proposed a point signature representation to describe the structural neighborhood of sample points. The point signature is defined as a distance profile circulated around the point and thus invariant to the part orientation. If the signatures of two points separately from the measurement and the part model are matched within the tolerance band, a correspondence pair is found. In addition, Hebert [8] introduced a spherical representation in which both the measurement and the part model are respectively represented by a spherical mesh. The mesh is computed from the data by deforming the standard shape objects. A surface curvature index is stored at every node of the mesh. The advantages of the spherical representation include comparing two meshes without combinatorial search and handling occlusions and partial views. Moreover, Yamany and Farag [9] presented a surface signature representation to convert the 3D surfaces into 2D surface images from certain points. The registration is performed by matching the generated surface images. Sun and Abidi [10] discussed a surface matching method based on the surface fingerprint that is defined by the 2D projection of geodesic circles on the tangent plane. On the other hand, given the popular applications of artificial neural network in feature recognition, the feed-forward network, the self-organizing feature map, and the Daugman's projection neural net have also been used in extracting the surface features from the measurements and matching to the part model [11–13].

The registration methods based on feature recognition work for any initial orientation. Once the features are extracted, the discrete matching process is straightforward, taking into account the relative small numbers of the features. An important factor affecting the performance of these methods is the density of the sample. The denser the sample point cloud is, the more accurately the features are extracted. Unfortunately, the dense point cloud is often acquired with the sacrifice of the sample quality. As such, the feature-based registration alone is not sufficient for precision inspection.

Providing a close initial pose, the part measurement is often refined in an iteration-based optimization process [14, 15], such as the iterative closest point (ICP) method [16]. Given a point cloud of measurement P and a nominal part model, the ICP algorithm works in three basic steps: (1) Compute the closest correspondence point p' on the part model for each measurement point p; (2) Calculate the transformation matrix T that minimizes the sum of square errors between the measurement point set and the correspondent set, i.e., $\sum_{i=1}^{n} \left\| p_i' - T \cdot p_1 \right\|^2 \Big/ n$; and (3) Apply the transformation matrix to repeat the first step until the objective value decreases less than the threshold value. In the second step, the closed-form solution with unit quaternions [17] can be used to acquire the rotation and transition vector of two correspondent sets, though the singular value decomposition is also possible [18]. Besl and McKay [16] proved that the ICP method will monotonically converge to the local minimum. In addition, they presented an extrapolation method to accelerate the iteration when four previous transformation vectors are well aligned. However, the most computationally intensive operation, the first step to calculate the closest points, was not addressed in detail.

Alternatively, Chen and Medioni [15] proposed to establish the correspondence pair by calculating the distance between the measurement point and the tangential plane of the correspondent point when two objects are close enough. While the registration is the same iteration-based process as in the ICP algorithm, there was no closed-form solution to calculate the transformation matrix of two correspondence sets. The standard optimization approaches such as conjugate direction method or quasi-Newton method would be used instead. Rusinkiewicz [19] reported that the point-to-plane correspondence has a faster convergence rate than the point-to-point correspondence, though the overall efficiency with the closed-form solution is not addressed.

Given monotonic convergence and straightforward implementation, many other variants have been proposed to improve the speed and robustness of the ICP algorithm. For instance, Masuda [20] introduced the least median of squares estimator to reduce the outlier effects and thus improve the registration robustness. Similar to the point-to-plane distance metric, no closed-form solution of two correspondence sets is available for the least median square metric. Alternatively, the closed-form solution could reject the worst 10% of correspondence pairs [21], the pairs inconsistent with neighbor distances [22], or the pairs on mesh boundaries [23] to improve the registration robustness. Based on the comparison study in [19], these approaches have similar effects on registration accuracy and stability by removing the outliers.

The most time-consuming operation of the iteration-based methods is to calculate the correspondence set. In addition to the discussed point-to-point distance and the point-to-plane distance, Blais [24] proposed finding the correspondence pixels of multiple views by reversing the calibration process of the rangefinder. Since these pixels are uniquely determined by their ray indices, the correspondence point is directly established by finding the pixel with the same ray index in other images. Note that the pixels with the same ray index in two views do not necessarily represent the closest point. This method provides a very fast way to identify the correspondence pixels in two images as it does not require the enumeration of all other pixels. However, this method is directly associated with the measurement generated by the image-based range sensors. In the case of the point-based laser probe, a k-d tree data structure is typically used to find the closest point. Furthermore, the k-d tree data structure can be specifically tailored to determine the closest points in the ICP method. For instance, Simon [25] used a data cache to save the five closest points in the last iteration. The next iteration would only use the cached five points. It is

interesting to note that there was no significant computation reduction (less than 10%) in the implementation of the five-point-cache scheme. Another nearest neighbor method is proposed by Greenspan and Godin [26]. In this method, the correspondences are tracked across iterations. If the distance estimation with previous correspondence satisfies the spherical constraints, then the k-d tree search is contained within the limited neighborhood points. Greenspan reported that the computation reduction became significant after 20 iterations.

The registration can match not only the geometry but also the detected intensity and colors. For instance, Godin [27] proposed the iterative closest compatible point method where the distance calculation is limited to the pairs with the compatibility index higher than the threshold value. Since the set of compatible points must be recomputed at each iteration, this will be a time-consuming registration process, slower than the standard ICP process.

In summary, although the ICP method is sensitive to the initial orientation, it is one of the methods that by far provide the most accurate registration result. A fair initial orientation is often available in the applications of precision inspection. In addition, the ICP algorithm is easy to be implemented with many possible speed and robustness enhancements. The next section will present an efficient implementation of the ICP method.

3 Efficient implementation of ICP registration

Based on the review of different methods, we have constructed a fast ICP algorithm by combining various techniques. Note that both Simon [25] and Rusinkiewicz [19] have presented high-speed ICP implementation for registering multiple views of range sensors. While these implementations have provided good case studies in comparing various ICP techniques, the goal of their methods was to rebuild the graphic objects based on the 3D images from the triangulation-based sensors. The registration process has focused on aligning two or more point sets together. The registration problem in precision inspection is a little different as it emphasizes aligning the measurement point set with the part model as accurate as possible. It may be argued that the part model can be represented by a discrete point set. But that would not be as accurate as the registration with the part model. As such, the techniques are different in the registration of point set and part model. For instance, Rusinkiewicz's implementation has significantly benefited from the projection-based correspondence based on [24], which is not applicable in the registration of point set and part model.

With these in mind, we will discuss an efficient ICP algorithm to register point clouds with the part model. Like many approaches, the closest point is computed on the approximated mesh model instead of the surface model. An approximated nearest neighbor data structure [28] was adopted to save the nodes of the part mesh model. The closest facet of each measurement point is cached across iterations. The closest distance is acquired first from the cached facet and its direct neighbors. If the correspondence pair is not perpendicular to the currently closest facet, a global search is initiated based on the approximated nearest neighbor representation. In addition, the iterations are accelerated in a similar scheme as [16]. These steps are further discussed in the following sections.

3.1 Compute the closest point

The free-form surfaces are typically represented as NURBS or other analytical forms. The most accurate distance is obviously the distance based on the analytical solution. But the analytical solution is a fairly sophisticated and slow optimization process. The common practice is to approximate the surfaces with the facet models. Investigation of the optimal facet approximation is beyond the scope of this paper. Our implementation has adopted a standard faceting module in the ACIS solid modeler. The distance of the measurement points to the surface model is then approximated by the distance to the facet model.

Once the facet model is available, a brute-force search that calculates the distances of the query point to all facets is possible but unacceptably slow when the number of facets is significant. Another alternative is to first identify the closest node to the query point, then compute the distance of the query point to the neighbor facets. It is very important to note that the closest node and its direct neighbor facets do not necessarily provide the closest distance. Figure 3 shows such an example. The distance of the point A to the mesh model is zero. But the calculation based on the closest node provided the wrong distance as indicated by the red line. Similarly was the point B. These wrong distances would affect the registration accuracy as well as the convergence rate. Thus it is necessary to build a topological facet representation to search beyond the first level of the neighborhood of the starting node.

With a k-d tree data structure, it takes $O(\log n)$ time to identify the closest node to any query point $\boldsymbol{q}$, in which n is the number of nodes in the part facet model. The symbolic representation has hidden the constant factor of computation time. When the number of nodes is large, it is still a significant operation. Moreover, Fig. 3 shows that even the closest node may not connect to the closest facet. On the

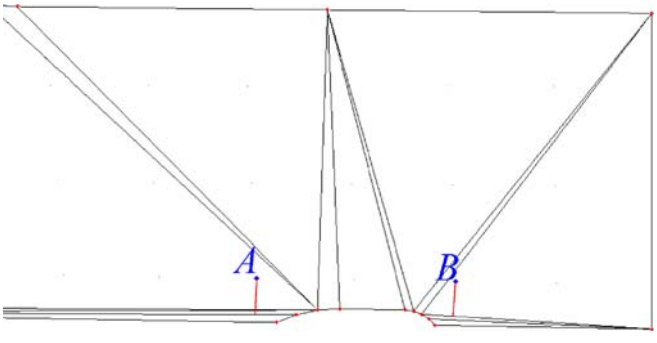


Fig. 3 The closest facet not connected to the closest node

other hand, Arya and Mount [28] have shown that if a small error is allowed (i.e., the returned node may not be the true closest node but is close to the true node within the error band), it is possible to significantly reduce the search time by using a balanced box-decomposition (bbd) tree. Assuming the true closest node is $\boldsymbol{p}^*$, the approximate nearest node $\boldsymbol{p}$ has the distance $d(\boldsymbol{p}, \boldsymbol{q})$ as:

$$d(p, q) \leq (1 + \varepsilon)\, d(p^*, q) \tag{1}$$

where $\boldsymbol{\varepsilon}$ is the relative error of the approximate nearest distance with respect to the true nearest distance. The search for the approximate closest point begins by locating the leaf cell that contains the query point $\boldsymbol{q}$. Next, the leaf cells are enumerated in increasing order of the distance to the query point. Denote $\boldsymbol{p}$ as the closest point enumerated so far. If the current leaf cell exceeds the distance $d(p, q)/(1 + \varepsilon)$, the search is terminated and returns $\boldsymbol{p}$ as an approximated nearest point to $\boldsymbol{q}$.

Once the approximate closest node is identified, the computation continues to find the closest facet on the part model. As discussed previously, the facets connected with the closest node do not necessarily include the closest facet to the calculated point. It is necessary to track other facets as well. Our implementation adopted a simple data structure in which the node, facet, and mesh are respectively defined as in Table 1.

Like any other representations, our data structure requires a one-time setup to build the topological neighborhood connection before the registration process. But the major operation during the ICP iterations, the computation of the closest point, has benefited from the immediate neighborhood information. Starting from the approximate nearest node, the search process goes through the neighbor facets in the direction of the least distance until a local minimum distance is found. The local minimum here is defined when no neighbor facets are closer to the query point, or the projected correspondence point is inside the traversed facet. The latter definition requires less computation because it avoids the additional neighbor checking.

The closest point computation can also be accelerated with a caching technique across iterations. Rather than caching the closest points [25], we store the closest facet of each point in $O(n)$ space. The search of the next ICP iteration starts with the cached facet instead of computing the closest node. If the cached facet or its neighbor is the local minimum for the query point, the closest distance is returned and the cache is refreshed. As the registered point cloud gets closer to the part model, the closest point often falls inside the cached facet or its neighbors that saved the computation of approximate nearest node. Note that the point cloud may not be close to the part model during the early stages of ICP iterations. It is not recommended to completely rely on the cached facet to compute the closest point. In our implementation, the computation switches back to the approximate nearest node search if the cached facet and its immediate neighbors do not provide the local minimum distance. As such, many incorrect local minimum distances are avoided, especially in the early iterations.

With the techniques of approximate nearest node and cached facets, the computation time of the closest points is significantly reduced. All samples in the point cloud can be taken into account to achieve the maximum accuracy at a fast convergence rate. To further improve the robustness of the registration process, the correspondence pair is considered outliers if the closest distance is more than three standard deviation away from the average closest distance. The correspondence pair with the closest point locating on the surface edge requires more caution. While the removal of these pairs may speed up the convergence rate as indicated in [19], it is also possible to remove the correct information from the registration process. An exaggerated case is the registration of a misaligned flat face. The disregarded point outside the surface may significantly affect the final registration accuracy.

Table 1 A simple facet model for efficient neighborhood search

```
Class CNode {
 CTypedPtrList<CObList, CNode *> NeighborNodes;
 CTypedPtrList<CObList, CFacet *> NeighborFacets;
};
Class CFacet {
 CTypedPtrList<CObList, CNode *> Nodes; // nodes in CCW
};
Class CMesh {
 CTypedPtrArray<CObArray, CNode *> MeshNodes;
 CTypedPtrArray<CObArray, CFacet *> MeshFacets;
};
```

3.2 Transform the correspondence set with signal-noise-ratio

The typical methods to register two correspondence sets include the unit quaternion [17] and singular value decomposition [18]. The unit quaternion method is also integrated with the linear vector acceleration in [16]. Our registration method adopts the unit quaternion with an additional weight factor derived from the sensor's signal-to-noise ratio.

The point-based sensor in our inspection system is a laser sensor based on conoscopic holography. The detected distance is proportional to the pattern frequency of equidistant linear fringes proportional in the conoscopic figure [29]. Any influence on the fringe frequency will also affect the measurement accuracy, e.g., target surface finish, laser incidence angle, standoff distance, etc. The signal-to-noise ratio (SNR) of the measurement represents the overlap of the fringe frequency from received laser beams. The 100% SNR means all received laser beams have generated exactly the same fringe pattern. The SNR value accompanies each individual point, and provides a confidence value by which weights can be assigned. As

such, the root of mean square error of the registration process is:

$$RMS(T) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} SNR_i \cdot \|p_i - Q(q_i)\|^2} \tag{2}$$

Where, n is the number of the correspondent pairs, $\boldsymbol{p}_i$ is the correspondence closest point of the measurement $\boldsymbol{q}_i$ on the design model with the signal-to-noise ratio SNR_i. $Q = \{Q^R, Q^T\}$ is the 7-element transformation vector, the first four of which Q^R is the rotation unit quaternion and the last three elements Q^T as the transition vector, $Q(q_i) = Q^R \otimes q_i - Q^T$. . In addition, the root square can be removed to facilitate the optimization process. As such, the objective is to minimize the following function:

$$\sum_{i=1}^{n} SNR_i \cdot \left\|p_i - Q^R \otimes q_i - Q^T\right\|^2 \tag{3}$$

The centroid of the measurement point cloud with the signal-to-noise ratio is:

$$\overline{p} = \sum_{i=1}^{n} SNR_i \cdot p_i \Big/ \sum_{i=1}^{n} SNR_i$$

Similarly,

$$\overline{q} = \sum_{i=1}^{n} SNR_i \cdot q_i \Big/ \sum_{i=1}^{n} SNR_i$$

Denote the new coordinates based on the centroids as

$$p_i^{'} = p_i - \overline{p},\ q_i^{'} = q_i - \overline{q}$$

Then the objective function can be written as

$$\sum_{i=1}^{n} SNR_i \cdot \left\|p_i^{'} - Q^R \otimes q_i^{'} - Q^{T'}\right\|^2$$

or

$$\sum_{i=1}^{n} SNR_i \cdot \left\|p_i^{'} - Q^R \otimes q_i^{'}\right\|^2 - \sum_{i=1}^{n} 2SNR_i \cdot \left\|p_i^{'} - Q^R \otimes q_i^{'}\right\| + \left\|Q^{T'}\right\|^2 \cdot \sum_{i=1}^{n} SNR_i \tag{4}$$

Since

$$\sum_{i=1}^{n} SNR_i \cdot p_i^{'} = 0, \quad \sum_{i=1}^{n} SNR_i \cdot q_i^{'} = 0,$$

the second item of Eq. 4 is zero. If the transition vector is the difference of two centroids, i.e., $Q^T = \overline{p} - Q^R \otimes \overline{q}$, then $\|Q^{T'}\|^2 = 0$. Furthermore, according to [17], the sum of square errors is minimized when the rotation quaternion Q^R is the eigenvector corresponding to the most positive eigenvalue of the symmetric matrix:

$$\begin{bmatrix} S_{xx}+S_{yy}+S_{zz} & \cdot & \cdot & \cdot \\ S_{yz}-S_{zy} & S_{xx}-S_{yy}-S_{zz} & \cdot & \cdot \\ S_{zx}-S_{xz} & S_{xy}+S_{yx} & -S_{xx}+S_{yy}-S_{zz} & \cdot \\ S_{xy}-S_{yx} & S_{zx}+S_{xz} & S_{yz}+S_{zy} & -S_{xx}-S_{yy}+S_{zz} \end{bmatrix}$$

where $S_{xx} = \sum_{i=1}^{n} SNR_i\left(p_i^x - \overline{q}^x\right)\left(p_i^x - \overline{q}^x\right)$, $S_{xy} = \sum_{i=1}^{n} SNR_i\left(p_i^x - \overline{q}^x\right)\left(p^y - \overline{q}^y\right)$, and so on.

3.3 Modified convergence acceleration

The ICP registration algorithm is an iteration-based process where the point cloud is brought close to the design model step by step until the step size ΔRMS is small enough. Depending on the initial pose and termination threshold, most of the registration process requires more than 50 iterations. Although non-derivative optimization methods (e.g., Powell-Brent method) can reduce the number of iterations, each iteration would need many more operations of objective function evaluation. Given the fact that the objective function evaluation is the most dominant operation, these methods would indeed increase the overall algorithm execution time. Besl and McKay [16] have presented an acceleration method in which the registration vector is extrapolated with linear or parabolic fit. This method is adopted in our implementation with simplified steps for the acceleration length, as shown in this section.

Given three continuous registration vectors $Q1$, $Q2$, and $Q3$, the direction difference in 7-d vector space can be defined by:

$$\theta = \cos^{-1}\left(\frac{(Q2-Q1)\cdot(Q3-Q2)}{\|Q2-Q1\|\cdot\|Q3-Q2\|}\right)$$

If θ is small (e.g., less than 10° as suggested in [16]), a new vector $Q3'$ can be used in the next iteration:

$$Q3' = Q3 + v * \frac{(Q3-Q2)}{\|Q3-Q2\|}$$

where v is the acceleration length, decided by linear or quadratic extrapolation. Suppose $r1$, $r2$, and $r3$ are the RMS values of $Q1$, $Q2$, and $Q3$, respectively. Let $v1 = -\|Q3-Q2\| - \|Q2-Q1\|$, $v2 = -\|Q3-Q2\|$, $v3 = 0$, then the optimal acceleration length according to the least square linear approximation is:

$$v_{min\_li} = -\frac{S_{rv}/S_{vv}}{\overline{r} - \overline{v}\cdot S_{rv}/S_{vv}} \tag{5}$$

where $\overline{r} = \frac{r1+r2+r3}{3}$, $\overline{v} = \frac{v1+v2+v3}{3}$,

$$S_{rv} = \sum_{i=1}^{3}(r_i-\overline{r})(v_i-\overline{v}),\ S_{vv} = \sum_{i=1}^{3}(v_i-\overline{v})^2$$

On the other hand, the optimal acceleration length according to the quadratic extrapolation is:

$$v_{min\_qua} = v2 - \frac{1}{2} \times \frac{(v2-v1)^2(r2-r3)-(v2-v3)^2(r2-r1)}{(v2-v1)(r2-r3)-(v2-v3)(r2-r1)} \tag{6}$$

The overall optimal acceleration length is

$$v_{\min} = \begin{cases} v_{min\_li} & \text{if } v_{min\_qua} \le 0 \\ Min\left(v_{min\_li}, v_{min\_qua}\right) & Otherwise \end{cases} \tag{7}$$

Compared to the original acceleration method [16], the presented approach requires three previous vectors well aligned on the same direction instead of four vectors. The selection rule of the acceleration step is much simpler than the original method. In order to prevent the overshoot, $v_{\min}$ is always less than 15 times the vector length $\|Q3-Q2\|$. Note this sanity length is smaller than the value in [16], but close to the value in [19]. In addition, Simon [25] reported to achieve faster convergence rate by separating the rotation and translation vectors. This was not the case in our experience. Because the sum of square errors is directly related to all seven parameters in the rotation and translation vectors, the acceleration of partial parameters (rotation only, translation only, or rotation and translation separately) could not provide a good objective function value in our applications. The rotation and translation vectors were indeed in different scale affecting the sum of square errors. But our test of normalized acceleration provides little differences on the convergence rate and registration accuracy. Additional details on the presented acceleration method are present in [30].

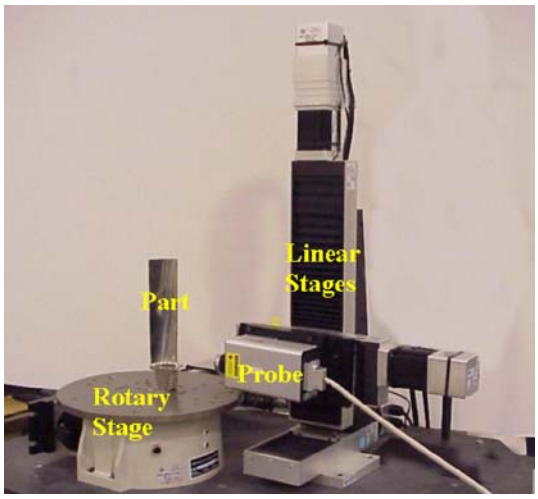


Fig. 4 Four-axis inspection machine

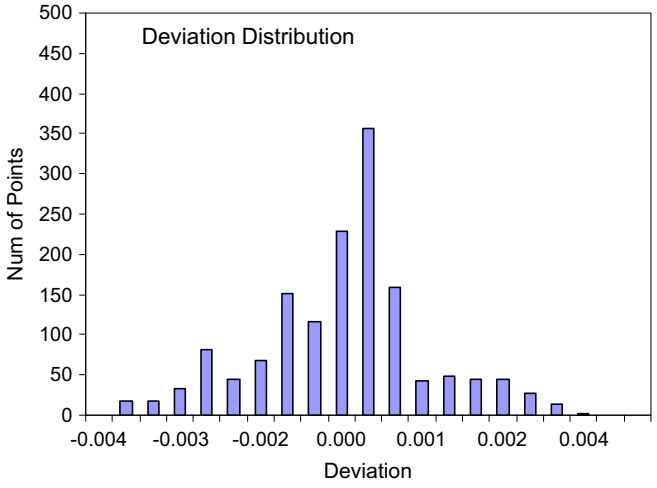
4 Registration in a four-axis inspection machine

A four-axis inspection machine is presented in Fig. 4. Three linear stages are adopted to provide x, y, z movements. The laser sensor is attached to the linear stage system. The part is placed on a rotation stage that can rotate 360 degrees. The data from the sensor and the stages are collected via the controller to provide the 3D position of measurement points. Aiming at the part family of turbine blades, the system is named the blade inspection machine (BIM) as one critical component of next-generation reconfigurable manufacturing systems that provides cost-effective comprise between production capability and functionality in response to rapid market change [31].

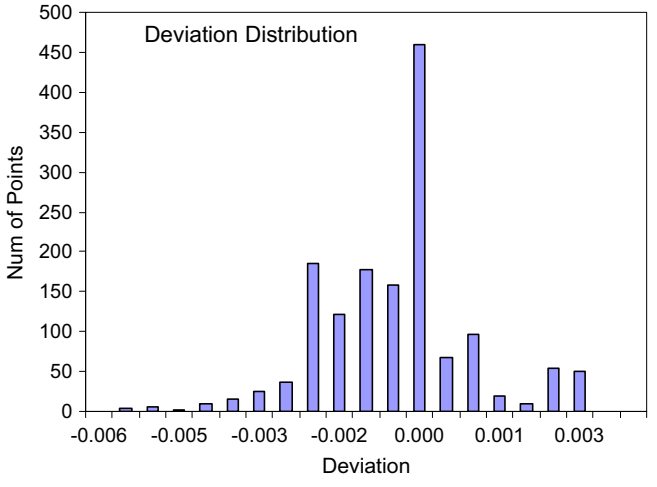
The BIM employed the point-based Optimet sensor (Optimet, Israel). The sensor probe can be reconfigured with different lens at different accuracy and working ranges. For instance, the 25 mm lens offers 3 μm measurement accuracy and 0.7 mm working range; and the 100 mm lens has less than 35 μm measurement accuracy but 70 mm working range. The larger working range will allow more flexibility in planning the probe path. The Optimet sensor also provides the signal-to-noise ratio (SNR) as the quality metric of measurement. A filter may be implemented to filter out the measurements with low SNR values (e.g., less than 70%). In addition, the sensor has wide incidence allowance over reflective or textured surfaces.

Note that the presented system does not necessarily include the fixture as the part may be placed on the rotary stage held by its own weight. For each view at the specified rotation angle, the non-contact measurements are collected and automatically aligned with the part model at the minimized root of mean square errors. In the applications of reverse engineering, the multiple views are first registered together to compare the part model with a complete measurement [32]. However, the registration of multiple views would introduce additional computation errors due to the discrete point representation. Therefore, in our application, each view is registered directly to the CAD model. Although it is argued that the registration of separate views and part model may underestimate the part error, this underestimation effect is negligible with carefully planned view positions. In addition, the registration of point set and part model can be based on the best fit of selected surfaces. As such, the registration of separate views and part model can guarantee the almost-zero-error alignments.

In order to assess the error of registration algorithm, a simulation program was established to mimic the part measurements with inspection errors. During simulation, the positions of the part, the sensor probe, the linear stages, and the rotary stage are characterized by Gaussian distributions as well as the sensor detections. As the measurements are simulated from the intersecting hit of the virtual laser ray and the digital part surface, the inspection error is determined by the mean value and standard deviation of these Gaussian distributions. For instance, if the positions had zero standard deviation, the inspection would have had no error at all. While this is not possible in the actual inspection process where the acquired measurement includes the part manufacturing error, the sensor detection error, the stage position error, and the fixture position error, the simulation provides a virtual test-bed environment to generate the measurement point cloud with controllable errors. These simulated measurements can then be used to verify the accuracy and effectiveness of automatic registration methods.

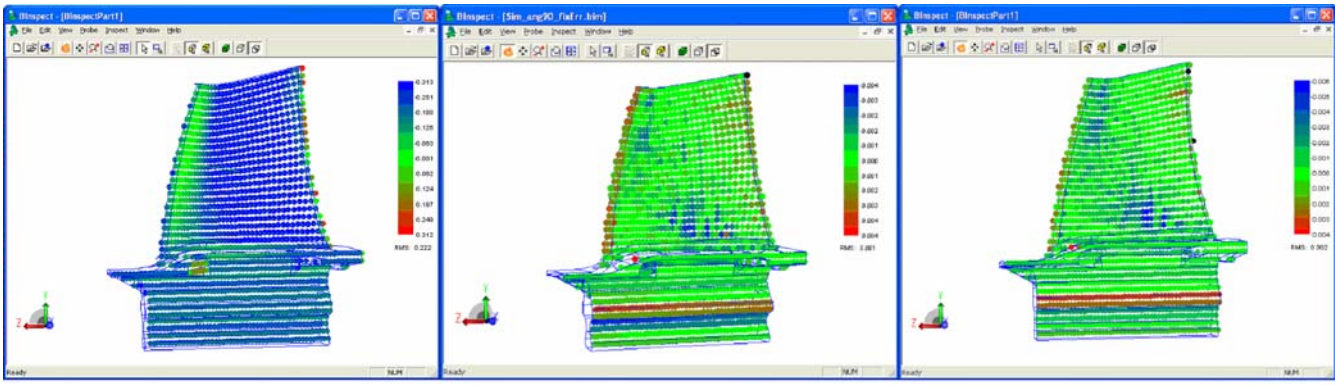


a Registered Measurements



b "Perfect" Measurements

Fig. 6 Deviation distribution. **a** Registered measurements. **b** "Perfect" measurements



a b c

Fig. 5 Registration with simulated measurements. **a** Misaligned measurements. **b** Registered measurements. **c** "Perfect" measurements

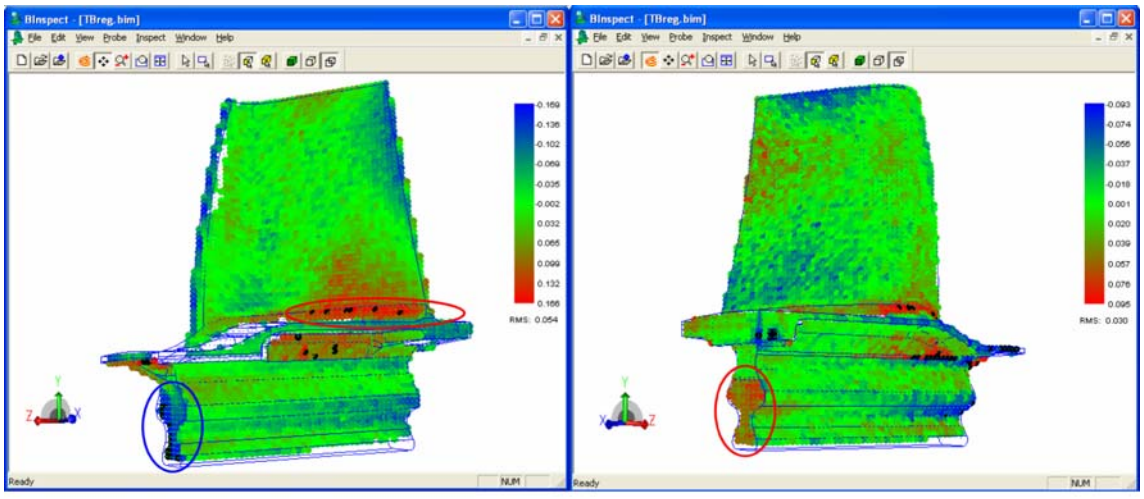


Fig. 7 Turbine blade inspection. **a** Registered front view. **b** Registered back view

Assuming a perfectly manufactured turbine blade with the fixture position error, the simulated measurement was misaligned with the part model as shown in Fig. 5a. The points were colored in different grades (from blue, green to red) to reveal the deviated distance. The sign of the distance indicated the location (inside or outside) of the measurement point with respect to the surface. The overall root of mean square errors (RMS) of the measurements was 0.222 mm due to the fixture position error in Fig. 5a. Then the registration was applied to automatically align the measurements to the part model. The RMS of the simulated measurements reduced to 0.001 mm after the registration with the minimum and maximum deviation at −0.004 mm and 0.004 mm, respectively, as shown in Fig. 5b. According to the previous discussion, the registration includes the error of facet approximation. This error could be evaluated by simulating perfect measurements (without any fixture or other errors) on a perfect part. Such a simulated measurement would not need any registration. Figure 5c shows that this "perfect" measurement has 0.002 mm RMS error, with the minimum and maximum deviation at −0.006 mm and 0.004 mm, respectively. Compared to this "perfect" measurement, the registered measurement actually has smaller RMS value. This is because the registration process automatically adjusts the deviation distribution to minimize the RMS value as indicated in Fig. 6, where the graph of Fig. 6a shows the normally distributed deviations of the registered measurement and Fig. 6b is the distribution of non-registered "perfect" measurement. In spite of this subtle difference, the colored deviation map of registered measurement in Fig. 5b provides a pattern consistent with the one of the "perfect" measurement in Fig. 5c. With this in mind, it could be concluded that the fixture position error has been compensated in Fig. 5b. The registration error due to the numerical iterations was nearly negligible.

The actual measurement was performed on the blade inspection machine. A preliminary transformation from the measurement coordinate system to the part coordinate

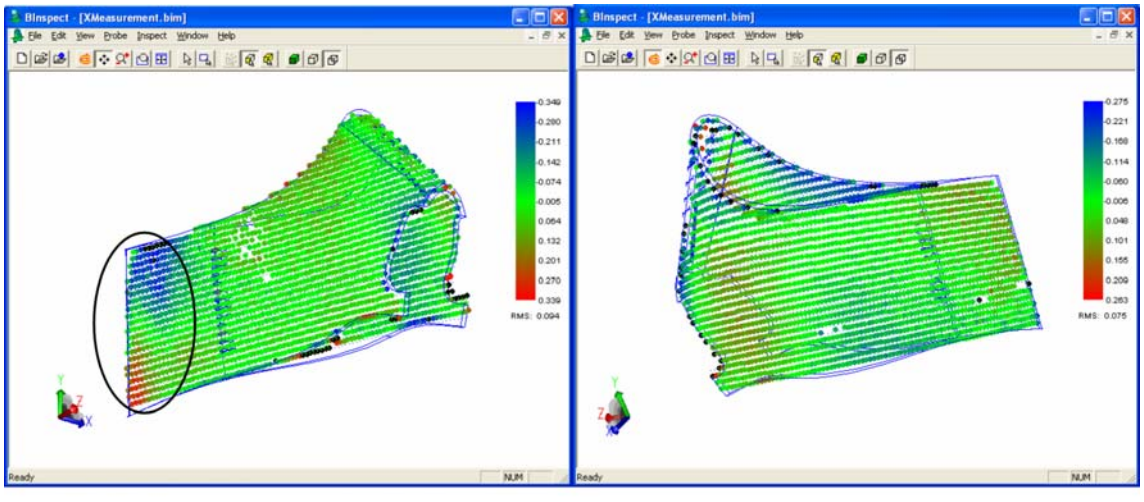


Fig. 8 Formed part inspection. **a** Registered front view. **b** Registered back view

Table 2 Efficiency of BIM registration compared to commercial package

Applications	Execution time (s)	RMS (mm)	Min distance (mm)	Max distance (mm)
Simulated measurements (102 surfaces, 1500 pts)	5 (30)	0.001 (0.009)	−0.004 (−0.038)	0.004 (0.035)
Turbine blade front view inspection (102 surfaces, 3354 pts)	6 (50)	0.054 (0.061)	−0.169 (−0.200)	0.166 (0.194)
Turbine blade back view inspection (102 surfaces, 3418 pts)	7 (45)	0.030 (0.030)	−0.930 (−0.930)	0.950 (0.970)
Formed part front view Inspection (165 surfaces, 2201 pts)	25 (200)	0.099 (0.103)	−0.365 (−0.365)	0.360 (0.358)
Formed part back view inspection (165 surfaces, 2209 pts)	31 (220)	0.074 (0.088)	−0.292 (−0.295)	0.280 (0.290)

system was acquired in the stage of machine setup. As we are heading to the low-precision-fixture or non-fixture inspection process, this preliminary transformation would not completely align the measurements with the part model. Instead, the model-based registration method could be applied to fine-tune the alignment of measurements, in which the preliminary transformation was used as the initial pose of the registration process. Figure 7 shows two registered measurements, the front and back views of a turbine blade. The part has the 0.1 mm surface tolerance. As shown by a number of highlighted dots with positive deviations in the figure, the bottom of the airfoil is found to be protruded from the nominal part shape. Furthermore, one side of the platform is away from the model (indicated by the positive deviations) and another side is inside the part (indicated by the negative deviations). This illustrates that the whole platform is shifted in relative to the airfoil position. It is worth noting that such a complete and precise inspection of free-form surfaces is very difficult and time-consuming for CMM inspection. The fixture-based industry practice that measures a few cross sections of the airfoil would not detect the part errors as shown in Fig. 7.

The blade inspection machine can also measure other parts with free-form surfaces and difficult fixture setup. Figure 8 depicts the registered measurements of a formed part. The front view shows that one side of the part is slightly twisted as the top is inside the part model (negative deviations) and the bottom is outside (positive deviations). Note that there are some black points in the colored deviation map. These are the outliers that are excluded from the RMS calculation. The outliers are defined as the inspection points with the absolute distance beyond three standard deviations of all measurements with respect to their signed distances.

The execution time and the deviated distance after registration are listed in Table 2. The test was performed on a home PC with 1.8 GHz Pentium4 CPU and 512 MB RAM. For the sake of comparison, the registration results from a commercial package are also listed in the brackets of the table. Note that both the BIM registration and the commercial package started at the same initial poses. The same three-sigma outlier filter was applied to the calculation of RMS, min, and max distances. In the tested applications, the BIM registration was found faster than the commercial package. The registration of simulated measurements revealed that the BIM registration could accurately bring the measurements close to the model while the commercial package could not.

5 Conclusion

This paper discussed an efficient registration method for the precision inspection of free-form surfaces. Different from the reverse engineering or graphics application where multiple views are registered together to provide a complete 3D model, the registration in precision inspection focuses on comparing the measurements with the part model at the maximum accuracy. The ICP algorithm is a popular method for this purpose. We have studied various techniques that accelerate the registration process and improve the efficiency of the ICP method. First, the combination of approximated nearest nodes and topological neighbor facets can significantly speed up each closest point calculation. The closest point calculation is further improved with the cached facets across the iteration steps. The registration method was enhanced by incorporating the signal-to-noise ratio into the transformation of correspondence sets. As such the outliers would have less or no effect in the weighted registration process. Last, an acceleration method based on linear or quadratic extrapolation can be fine-tuned to provide the fast yet robust iterative solution.

These techniques have been implemented in a four-axis blade inspection machine with no requirement of precision fixtures. The virtual simulation was also established to verify the registration correctness. These simulated measurements indicated that the implemented registration algorithm is fast and precise. A turbine blade part and a formed part are measured on the blade inspection machine. The registered measurement reveals part defects that are very difficult to detect with current industrial practices. With this promising progress, future work includes the test of a registration algorithm on more precision inspection applications.

Acknowledgement The authors thank Neil Craft from Williams International Co. for providing the physical parts and geometry models used in this paper.

References

1. Spyridi AJ, Requicha AAG (1990) Accessibility analysis for the automatic inspection of mechanical parts by coordinate measuring machines. IEEE international conference on robotics and automation, Cincinnati, Ohio
2. Brown CW (1991) IPPEX: an automated planning system for dimensional inspection. Manuf Syst 20(2):189–207
3. Hopp TH (1984) CAD-directed inspection. Ann CIRP 33

4. Blais F (2003) A review of 20 years of range sensor development. In: Videometrics VII, proceedings of SPIE-IS&T electronic imaging, Santa Clara, CA
5. Giusarma S, Moroni G, Polini W (2004) Inaccuracy prediction due to six-point locating principle. Proceedings of the CIRP ICME '04, 30 June – 2 July, Sorrento, Italy, pp 213–218
6. Barhak J, Djurdjanovic D, Spicer P, Katz R (2005) Integration of reconfigurable inspection with stream of variations methodology. Int J Mach Tools Manuf 45(4/5):407–419
7. Chua CS, Jarvis R (1997) Point signatures: a new representation for 3D object recognition. IJCV 25(1):63–85
8. Hebert M, Ikeuchi K, Delingette H (1995) A spherical representation for recognition of free-form surfaces. IEEE PAMI 17(7):681–690
9. Yamany SM, Farag AA (1999) Free-form surface registration using surface signatures. Proceedings of the IEEE international conference on computer vision, Kerkyra, Greece, pp 1098–1104
10. Sun Y, Paik JK, Koschan A, Page DL, Abidi MA (2003) Point fingerprint: a new 3-D object representation scheme. IEEE Trans Syst Man Cybern Part B 33(4):712–717
11. Ghosal S, Mehrotra R (1995) Range surface characterization and segmentation using neural networks. Pattern Recogn 28 (5):711–727
12. Knopf GK, Sangole A (2002) Registration of closed free-form surfaces using deformable maps. Proceedings of the artificial neural networks in engineering (ANNIE '92) conference, St. Louis, MO
13. Wu X, Li D (2003) Range image registration by neural network. Mach Graph Vis 12(2):257–266
14. Faugeras OD, Hebert M (1986) The representation, recognition, and locating of 3-D objects. Int J Robot Res 5(3):27–52
15. Chen Y, Medioni G (1992) Object modeling by registration of multiple range images. Image Vis Comput 10(3):145–155
16. Besl PJ, McKay ND (1992) A method for registration of 3-D shapes. IEEE PAMI 14(2):239–255
17. Horn BKP (1987) Closed-form solution of absolute orientation using unit quaternions. J Opt Soc Am A4(4):629–642
18. Arun K, Huang T, Bolstein S (1987) Least-squares fitting of two 3-D point sets. IEEE PAMI 9(5):698–700
19. Rusinkiewicz S, Levoy M (2001) Efficient variants of the ICP algorithms. Proceedings of the third international conference on 3D digital imaging and modeling, Quebec City, Canada
20. Masuda T, Yokoya N (1995) A robust method for registration and segmentation of multiple range images. Comp Vis Image Underst 61(3):295–307
21. Pulli K (1999) Multiview registration for large data sets. Proceedings of the international conference on 3D digital imaging and modeling, Ottawa, Canada, pp 160-168
22. Dorai C, Wang G, Jain AK, Mercer C (1998) Registration and integration of multiple object views for 3D model construction. IEEE PAMI 20(1):83–89
23. Turk G, Levoy M (1994) Zippered polygon meshes from range images. Proceedings of SIGGRAPH 94, Orlando, Florida, 24-29 July
24. Blais G, Levine MD (1995) Registering multiview range data to create 3D computer objects. IEEE PAMI 17(8):820–824
25. Simon DA, Hebert M, Kanade T (1994) Real-time 3-D pose estimation using a high-speed range sensor. Proceedings of the IEEE international conference on robotics and automation (ICRA '94), San Diego, CA, vol 3, pp 2235-2241
26. Greenspan M, Godin G (2001) A nearest neighbor method for efficient ICP. Proceedings of the third international conference on 3D digital imaging and modeling, Quebec City, Canada
27. Godin G, Rioux M, Baribeau R (1994) Three-dimensional registration using range and intensity information. SPIE Videometrics III 2350:279–290
28. Arya S, Mount DM, Netanyahu NS, Silverman R, Wu A (1998) An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. JACM 45(6):891–923
29. Sirat GY, Paz F, Kleinman M, Doherty M (2003) ConoProbe and conoline, two new 3-dimensonal measurement systems. LPM2002, proceedings of SPIE 4830:319–324
30. Zhu L, Barhak J, Srivatsan V, Katz R (2004) Error analysis and simulation for a four-axis optical inspection system. Proceedings of DET 2004, Seattle, WA
31. Koren Y, Ulsoy A (2002) Vision, principles and impact of reconfigurable manufacturing systems. Powertrain International, pp 14–21
32. Bergevin R, Soucy M, Gagnon H, Laurendeau D (1996) Towards a general multiview registration technique. IEEE PAMI 18(5):540–547