Int J Adv Manuf Technol (2010) 49:1093–1106
DOI 10.1007/s00170-009-2447-3

ORIGINAL ARTICLE

# Verification of 3D freeform parts by registration of multiscale shape descriptors

Alex Shmukler · Anath Fischer

Received: 27 August 2009 / Accepted: 12 November 2009 / Published online: 10 December 2009


**Abstract** Precision inspection of freeform parts takes an important role in manufacturing quality control. The aim of this inspection is to verify that the geometric dimensions and produced part tolerances meet quality requirements. This is achieved by fitting the scanned data to the computer-aided design (CAD) model. This verification is complicated since the produced part includes defects and distortions. Currently, industry uses semimanual verification, which is expensive, often inaccurate, and very time-consuming. This paper describes a new method for automatic registration and alignment of two 3D freeform shapes, one from the scanned data and the other from the CAD model. The method makes no assumptions about their initial positions. Instead, the proposed algorithm uses a multiscale shape descriptor to select features on the scanned data and identify their corresponding features on the CAD model. The proposed shape descriptor is invariant with respect to local shapes and is robust to noise. A coarse alignment is computed by finding and registering the best matching triplet of features. The iterative closest point algorithm uses resulting coarse alignment to achieve a tuned alignment. The proposed method is automatic, efficient, and straightforward to implement. The algorithm can also be effective in the case of partial scanned inspected shapes. The feasibility of the proposed method is demonstrated on a blade model.

**Keywords** Precision inspection · Verification · Freeform features · Registration

A. Shmukler · A. Fischer (✉)
CAD & LCE Laboratory,
Faculty of Mechanical Engineering Technion,
Haifa 32000, Israel
e-mail: meranath@technion.ac.il

## 1 Introduction

Precision inspection of freeform parts takes an important role in manufacturing quality control. The prevalent idea is to verify that the geometric dimensions and part tolerances meet the quality requirements of manufacturing. Currently, carry out this entire process. Manual inspection is expensive, often inaccurate, and very time-consuming. Speed, accuracy, and repeatability are the major challenges to be achieved by a portable and flexible inspection system. Ideally, such a system would automatically collect the measurements and make detailed, full-scale comparisons between the measured data and the original design. With this capability installed on machine tool and assembly lines, manufacturing processes can be monitored and controlled in real time. This is especially important in the era of globalization, where a production process may be distributed over several sites.

A typical inspection process consists of the following general phases: (1) data acquisition; (2) mesh generation; (3) registration of the measured data with the designed model; (4) verification. 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 designed part coordinate system. Therefore, this work focuses on data alignment for defect detection (phase c).

### 1.1 Data acquisition

Data acquisition is applied by using a device capable of obtaining 3D geometry data from the inspected object. Common devices are [1]: (1) laser-based scanners; (2) coordinate-measuring machine (CMM) which is essentially a Cartesian robot with one tactile probe; (3) stereoscopic cameras; (4) acoustic devices; and (5) medical-scanning devices (magnetic resonance imaging, computed tomography). As of today, CMMs is the most widely used device for industrial inspection. While the CMM measures the parts in high precision, a conventional tactile probe is often limited in scanning speed and cannot cover features that are smaller than the stylus diameter. The tactile probe is especially cumbersome in measuring parts with freeform surfaces as it is difficult to maintain continuous contact with the surfaces. In these cases, noncontact scanners such as laser scanners are commonly used for fast surface acquisition. Usually, a triangular mesh is reconstructed from the sampled points.

### *1.1.1 Discrete curvature of meshes*

A large range of problems in 3D computer vision and computational geometry can be solved by means of surface curvature properties. Curvature is a differential parameter that is also used in registration algorithms because it represents properties that are invariant under Euclidian transformations, and thus can be used for characterizing a surface [2, 3]. The most common geometric surface representation in RE is triangular mesh, constructed from sampled points. It has $C^1$ zero curvature at faces and $C^0$ continuity at vertices and edges. However, the mesh is approximation of a real unknown piecewise smooth surface. Therefore, the surface curvature can be estimated from the triangular mesh and be represented as discrete Gaussian curvature $K$ and the absolute discrete mean curvature $|H|$ [4]. Based on these discrete curvatures, features can be extracted. Most algorithms calculate the vertex curvature in the region of one-ring neighbors of a vertex. These algorithms are very noise sensitive and are topological dependent [5–7]. These algorithms give good results for synthetic objects and had problems with noisy scanned objects. Hamann and Taubin [8, 9] improve the methods by applying their algorithm on the $n$-ring neighborhood.

The Gauss–Bonnet theorem can be applied to triangular meshes [10], where the local region M can be defined as a one-ring neighborhood of a given vertex:

$$\int\int_M K dA = 2\pi - \sum_{j=1}^{\mathrm{nf}} \theta_j \tag{1}$$

where $\theta_j$ is the angle of the $j$th face at the given vertex, and nf denotes the number of faces around this vertex. The angular deficit is analog to the curvature, it measures how curved a vertex is. In order to derive a Gaussian curvature at a given vertex, it is assumed that the curvature is uniformly distributed around the vertex, and is normalized by the area $A$:

$$K = \frac{2\pi - \sum_{j=1}^{nf} \theta_j}{\frac{1}{3}A} \tag{2}$$

where $A$ is the sum of the faces area around the given vertex.

Mean curvature can also be formulated as a concept of an angle:

$$H = \frac{\frac{1}{4}\sum_{j=1}^{ne} \left\| v_j - v_i \right\| \gamma_j}{\frac{1}{3}A} \tag{3}$$

where $\gamma_j$ is the dihedral angle between two faces of edge $(v_i, v_j)$,see Fig. 1.

Meyer et al. [11] suggested a modified formula for mean curvature based on simplification of the integral of the Laplace–Beltrami operator:

$$H = 0.5 \cdot \left\| \frac{1}{2A_{\mathrm{vor}}} \sum_{j=1}^{\#e} \left( cot\alpha_j + cot\beta_j \right) \left( v_j - v_i \right) \right\| \tag{4}$$

where $v_j$ is a neighbor of $v_i$, and $\alpha_j$ and $\beta_j$ are the two angles opposite to the edge $(v_i, v_j)$ of two triangles. Where $A_{vor}$ is defined as the Voronoi area of vertex $v_i$, when the triangles are nonobtuse:

$$A_{\mathrm{vor}} = \frac{1}{8} \sum_{j=1}^{ne} \left( \cot\alpha_j + \cot\beta_j \right) \left\| v_j - v_i \right\|^2 \tag{5}$$

In our approach, the discrete curvature estimation is based on the Gauss–Bonnet scheme. In order to improve the robustness of the method, we apply it on $n$-ring neighborhood. The advantage of these techniques is that they are very fast, since they do not require normal estimation.

## 1.2 Registration methods

Registration of two objects in arbitrary positions is a fundamental problem in object acquisition and modeling. Existing geometric registration methods solve the problem

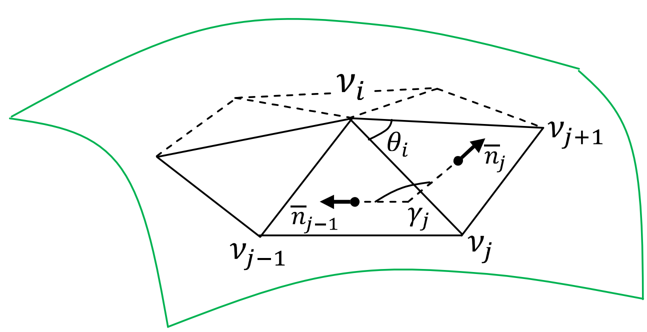



**Fig. 1** A vertex and its one-ring neighbors

of best shape alignment by matching the correspondence parameters between two object models. It is divided in two subproblems: correspondence and alignment. The input sets are usually the scanned (noisy) model and the computer-aided design (CAD) model of the object (Fig. 2). The correspondence parameters can be the object's sampled points, or can be object's geometric features such as holes, edges, and corners. For freeform parts that do not have standard geometric features, the correspondence parameters are more complex. Existing registration methods can be classified into two categories, global and local registrations [12]. Global registration methods align the models, without any prior assumptions about their initial positions. These algorithms are usually based on a correspondence search. Local registration algorithms assume that the relative transformation between the models is small, and improve the registration locally. These methods are almost exclusively based on distance local optimization.

### 1.2.1 Global registration methods

Global registration algorithms make use of the fact that the rigid transform is low dimensional, i.e., needs only a small number of correspondence parameters to define the optimal transformation. One class of global registration is known as voting methods [13], quantizes the transformation space into a 6D matrix, and accumulates votes in each matrix cell. The entry with the most votes gives the optimal aligning transform. These methods compute a transformation that aligns triplet points. Voting methods explore the entire set of transformations and therefore are likely to find the optimal alignment. However, these methods have high computational complexity of at least $O(n^3)$, and therefore are not used commonly for global registration. A second class of global registration approaches is based on analysis

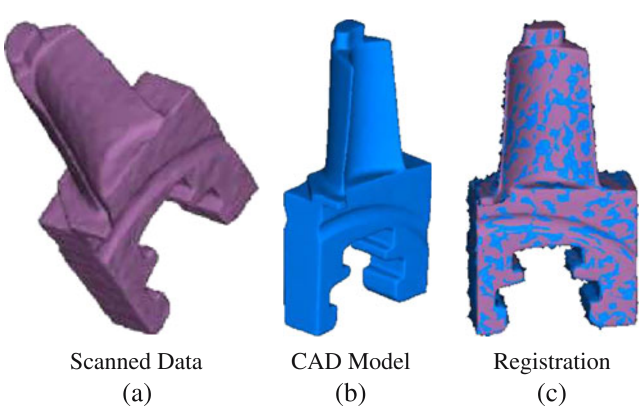



**Fig. 2** The registration process. **a** Scanned data model in local coordinate system. **b** CAD model in computer coordinate system. **c** The registered scanned and CAD models in global coordinate system

of the error function of Eq. 14. Since minimization over all transformations is a nonlinear function, it may contain several local minima. Therefore, in addition, methods that eliminate local minima have been applied. However, these methods are slow and are not always converge. The third class of approaches is known as correspondence search methods. Given a set of corresponding point-pairs, a rigid transform is computed so that the distance between corresponding points is minimized [14]. Both the voting schemes and the correspondence search algorithms can be improved by using geometric descriptors.

### 1.2.2 Local registration methods

Local registration methods get as input a rough estimation of aligning transformation between the scanned and the CAD models. The local registration algorithm then refines this rough alignment. Most local registration algorithms are variations of the iterative closest point (ICP) algorithm first introduced by [15]. The ICP algorithm converges to a local minimum of Eq. 5. It has been shown both experimentally and theoretically that the convergence behavior of ICP depend heavily on the choice of corresponding points and the predefined distance metric. The main limitation of ICP and its variants is that, it is not effective when the relative initial position between the input models is not close. Using geometric descriptors as a correspondence features can speed up the convergence significantly.

## 1.3 Geometric shape descriptors

A competent speedup technique for registration is to find a set of feature points on the models based on computed geometric descriptors. A geometric shape descriptor is a quantity computed for each point of the models, based on the shape of its local neighborhood. High-dimensional descriptors provide a detailed description of the shape around each point. The advantage of using geometric shape descriptors is that, given a point on the scanned model, it is likely that only a few points in the CAD model will have a similar descriptor. Moreover, the point with the best matching descriptor is likely to be the preferred corresponding point. Incorrect correspondences are few and can be removed using outlier detection methods. In global registration, geometric shape descriptors are often used to solve the correspondence problem directly. Low-dimensional descriptors are typically much easier to compute and compare than high-dimensional descriptors [16]. However, for a given point in the scanned model, there may be many points in the CAD model with the same descriptor value. Therefore, low-dimensional descriptors are usually used in conjunction with a voting scheme [17] to reduce the size of the search space. Most of the low-dimensional shape descriptors are based on differential quantities of the shape, since they are invariant under rigid transformations. Their main limitation is that they are sensitive to noise. As a result, the scanned model should be presmoothed. An alternative approach is to use local shape invariants that are based on integration instead of differentiation [18]. Integral descriptors have desirable properties of differential invariants of locality and invariance under rigid transformations, and they are more robust to noise.

In this research, a registration method is proposed, based on feature extraction and correspondence search using 3D integral geometric descriptors. Furthermore, our method combines both the global and the local registration for coarse and fine alignment accordingly.

#### *1.3.1 The verification analysis*

Registration is very time-consuming, and any inaccuracy of this algorithm results in erroneous verification analysis. Additionally, complicated mechanical parts can have many curved features, and the position or orientation of some features can severely deviate from its original model [19]. In such a case, the whole part fails the verification. In recent years, several publications on quality control have been based on invariant surface properties [20]. All these methods assume that the scanned object must first be smoothed; this will damage the surface geometry, and the curvature will be wrongly estimated. The verification of engineering parts can be improved by applying shape descriptor analysis between the corresponding models. The shape descriptor's robustness to noise enables to achieve accurate verification of noisy data.

## 2 The approach

In this paper, a method is described for calculating a rigid transformation that optimally aligns 3D sampled model with respect to the CAD model. The inspected scanned model is represented as a noisy triangular mesh, and the CAD model is represented as triangular mesh without noise. The main stages of the algorithm are as follows (Fig. 3):

1. Estimating the mean curvature of the scanned and CAD models, using an extended neighborhood method for mean curvature estimation
2. Computing the geometric shape descriptor of the scanned and CAD models
3. Matching between the scanned and CAD models
4. Applying registration of the scanned and CAD models

The stages are described in detail in the next sections.

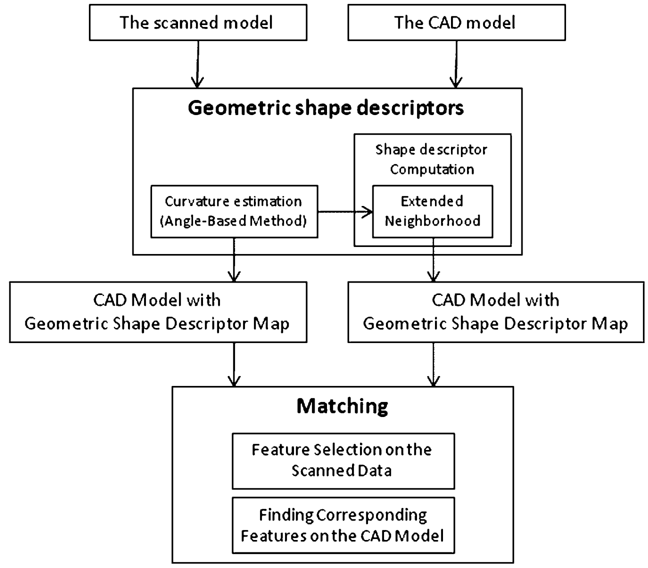



**Fig. 3** The main stages of the algorithm

### 2.1 Discrete mean curvature over one-ring neighborhood

The angle-based method for mean curvature estimation is fast and straightforward to implement. Furthermore, curvature can be computed for open surfaces. According to [5], it can be applied for regular meshes, but the algorithm is inaccurate for highly dense sampled objects. The mean curvature map of an object is shown in Fig. 4. The red regions indicate local peaks/pits. The blue regions indicate saddle shaped. The green regions indicate local flat shapes. The effect of noise can be eliminated by using the proposed multiscale shape descriptors.

### 2.2 Discrete mean curvature over $n$-ring neighborhood

Most curvature estimation methods relate to the first-ring neighborhood of the vertices. Our estimation method calculates the mean curvature over an extended neighborhood around a given vertex ($n$-ring neighborhood). It relies on the surface domain $D$ bounded by the integration kernel with radius $r$ ($n$-ring). First, the mean curvature is computed at each vertex, based on the one-ring connectivity, according to Eq. 2. Then, the local mean curvature $\widetilde{H}$ is estimated on the surface domain $D$ by applying the following weighted equation:

$$\widetilde{H}(v_i) = \sum_{j=1}^{n} \omega_{ij} \cdot H(v_j) \tag{6}$$

where:

- $\omega_{ij}$ is a weighted factor between the center vertex $v_i$ and neighbor $v_j$.

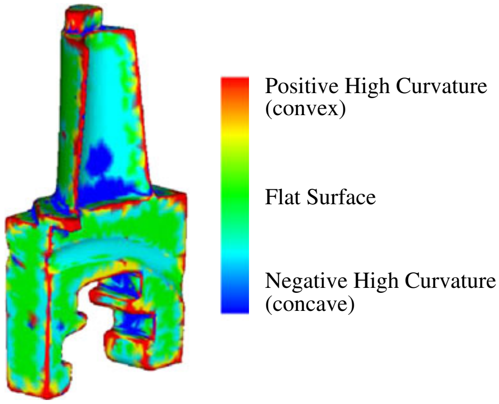



**Fig. 4** Curvature map

- $H(v_j)$ is a mean curvature of $v_j$ computed according to Eq. 3
- $n$ is the number of neighbor vertices contained in the domain $D$.

The weight factor $\omega_{ij}$ is depending on the geodesic distance [21] between vertices $v_i$ and $v_j$ (see Fig. 11). We have used the following exponential function which gave stable results:

$$\omega_{ij} = e^{-\frac{g_{ij}}{g_{i,\max}}} \tag{7}$$

where:

- $g_{ij}$ is a geodesic distance between vertices $v_i$ and $v_j$.
- $g_{i,max}$ is a max geodesic distance from vertex $v_i$ to the other vertices in $D$.

### 2.3 Multiscale geometric shape descriptor

As discussed above, the geometric methods for feature detection commonly used in registration algorithms are very sensitive to noise. Therefore, we chose a different approach based on a discrete scale parameter of the local neighborhood. This parameter is a multiscale geometric shape descriptor that makes it possible to deal with noisy data and to obtain invariant measurements on multiple scales, without any change in the given geometry, such as those caused by smoothing. A multiscale shape descriptor, described in Section 1.3, is a vector of scalars assigned to each point in the model. For robust registration algorithms, the descriptor should be invariant to rigid transformations, robust to noise, and based on the local geometry around any point of the model. These local neighborhoods are typically defined via balls or spheres, also called kernels, whose radius defines the working scale. This approach focuses on low-dimensional descriptors based on integral invariants, since they are faster to compute than rich descriptors, invariant under rigid transformations, and robust with respect to noise. Integral invariants are defined by integrating spatial functions over moving domains centered at surface points. Here, we briefly introduce the concept; a detailed treatment of the theory and computation can be found in [22].

Let $P$ be the input shape, consisting of $N$ points $p_1 \ldots p_N$. The integral volume descriptor is a 3D integral invariant that is defined at each vertex p of the input shape as follows:

$$V_r(p) = \int_{B_r(p)\cap D} dx = \int_{B_r(p)} \chi_D(x)dx \tag{8}$$

where:

- $Br(p)$ is the integration kernel that represents a ball of radius $r$ centered at the $p$.
- $D \subset \mathbb{R}^3$ is the boundary of the surface domain represented by input shape $P$.
- $\chi_D$ is the characteristic function which is 1 for points of $D$ and 0 elsewhere.

The quantity V$r(p)$ is the volume of the intersection of the ball $Br(p)$ with the interior of the object defined by the input mesh. Assuming that the intersection of $D$ and $Br(p)$ is simply connected, the integral volume descriptor is related to mean curvature at point $p$ as follows:

$$V_r(p) = \frac{2\pi}{3} r^3 - \frac{\pi H_r}{4} r^4 + O\left(r^5\right) \tag{9}$$

where:

- $H_r$ is a local mean curvature of a bounded domain $D$ at a point $p$.
- $r$ is a radius of a kernel $Br(p)$.

The leading (first) term is the volume of the half-ball of radius $r$, and the correction term involves the mean curvature $H_r$ at the point $p$. The proof that this descriptor is robust to noise can be found in [22]. In this work, we use a modified form of the integral volume descriptor, which characterizes the ratio between the volume of the intersection $Br(p)$∩D and the volume of the entire ball $Br(p)$. In other words, we normalize the magnitude of the volume descriptor $Vr(p)$ by the volume of the ball $\frac{4\pi}{3} r^3$:

$$V_r(p) = \frac{1}{2} - \frac{3}{16} H_r \cdot r + O\left(r^2\right) \tag{10}$$

Note that in the case of a planar surface, $Vr(p)=1/2$.

Integral descriptors are particularly suited for multiscale representation since the scale is controlled by the radius of the kernel $Br(p)$. As can be seen from Eq. 10, the modified integral volume descriptor is a linear function of the mean curvature $H_r$ multiplied by a kernel radius. Therefore, the volume descriptor can be computed at different scales. This multiscale descriptor is used in a registration algorithm for picking potential feature points on the input surfaces and identifying corresponding points for those features. A feature point $p$ can be detectable over a set of consecutive scales of the descriptor. For a false feature point $p$, the descriptor will not match in all scales. The volume descriptor on a 3D model is illustrated in Fig. 5. The multiscale descriptors are demonstrated in Figs. 9 and 12 (Section 4).

The integral volume descriptor can be computed using voxel grid techniques. The drawback of this method is that the grid distribution is highly dependent on mesh quality. The other technique is to directly compute the multiscale integral volume descriptor at each scale by using the relation of mean curvature $H_r$. The problem is to estimate the mean curvature of the bounded surface domain $D$, when the surface is represented as a triangular mesh. The proposed solution to the estimation problem is presented in the following section.

### 2.4 Feature selection

In most scanning applications, the input data is too large for the matching process. Therefore, a subset of points (features) is extracted from the scanned model, mainly from salient or unique areas of the scanned model. Then, a simple and fast registration algorithm can be applied on those features. In this work, feature selection is based on analyzing the distribution of geometric shape descriptors. The features are those vertices whose descriptor values are not common, over consecutive scales. The result is only a few corresponding points in the CAD model. It is sufficient, since a rigid transform can be determined by small number of points. The performance of the feature extraction method is depended on the sample density. For a denser sampled point cloud, the features are extracted more accurately. Unfortunately, a dense point cloud is often acquired at the expense of sample performance.

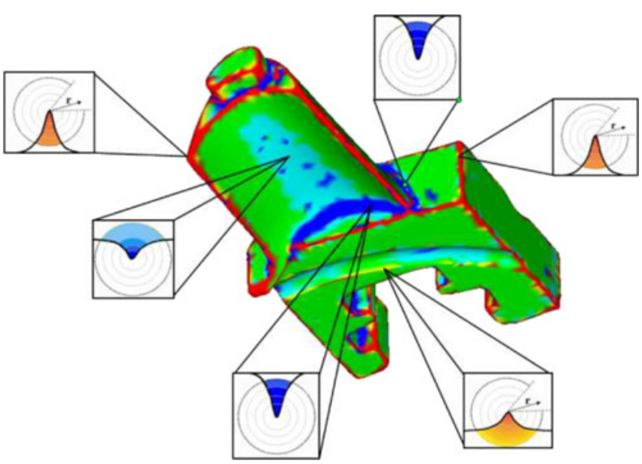


**Fig. 5** Mapping of an object according to the integral volume descriptor

### 2.5 Correspondence search

Once the feature points are extracted, the discrete matching process is straightforward. Given a set of features extracted from the scanned data, a set of best corresponding points are found on the CAD model. The matching is based on comparing the volume descriptor values over consecutive scales of both scanned and CAD models. Then, the optimal sub set of corresponding points are selected. The verification algorithm is based on a distance comparison between triplets of points from both models.

### 2.6 Registration

Once the optimal correspondence set is determined, the models can be aligned. Two types of pairwise registrations are applied, global and local alignment. *Global registration*: Starting from an arbitrary initial position of the scanned and CAD models, the coarse alignment is defined. The global registration algorithm is based on finding a transformation that best aligns two models. The transformation may be computed based only the best triplet of features selected from the scanned and CAD models. *Local registration*: After the coarse alignment, a small number of iterations of standard ICP with point-to-point error metric are applied, thus bringing the models into high accurate alignment. In addition, to achieve high accuracy and robustness to noise, the verification may be based on the volume descriptor analysis. Our algorithm is able to align fully and partially scanned parts with the CAD model, and is robust with respect to noisy data. The algorithm works well in the presence of strong point-like features. The implementation of the proposed approach is presented in the following section.

## 3 The implementation

In industrial applications, real-time data processing plays a crucial role. Therefore, implementation must be fast and efficient. This study has been implemented in C++ with the OpenGL graphic library.

### 3.1 The data structure

The data structure chosen for this research handles triangulated meshes. One of the requirements of this data structure is that it must be fast created, and the curvature algorithms must have direct access to the mesh vertices and their neighbors. The data structure that fulfills these requirements is the mesh data structure, composed of vertices and faces. In the input data, the connectivity between vertices and faces is not defined explicitly. Therefore, connectivity is determined. As a result, a one-ring neighborhood structure is created. This bidirectional connectivity is crucial to enable fast computation [23]. The presented data structure includes properties of mean curvature and multishape descriptor per vertex. However, this data structure has high space complexity, still can be handled in PC.

### 3.2 Mean curvature estimation over an $n$-ring neighborhood

The dihedral angle is defined as the angle between two faces that share a common edge. It is used in estimating the angle-based mean curvature (Section 2.1). The algorithm calculates the curvature in a one-ring neighborhood. is very fast; it runs in seconds for thousands of vertices and has linear time complexity of $O(m)$ per vertex, where $m$ is the number of neighbors. The $n$-ring neighborhood per vertex relies on the surface domain (see Section 2.2). Figure 6 shows the one-ring (green points) neighbors and vertices from the extended neighborhood (brown points) bounded by the current integration kernel ball (blue circle) around a given central vertex (red point). Calculating the mean curvature over the extended neighborhood consists of two stages: First, construction of an extended neighborhood structure is defined. Then, the mean curvature is calculated.

An extended neighborhood structure is a map of vertices around a given vertex which are located inside a given sphere with radius (kernel) $r_k$. Since the radius depends on the current scale of the integral volume descriptor (Section 2.3), a separate map at each scale $k$ is calculated. In other words, an extended neighborhood structure is a multimap with scale $k$. Each vertex in the map is set with the exact geodesic distance from the given vertex. The proposed algorithm for building an extended neighborhood

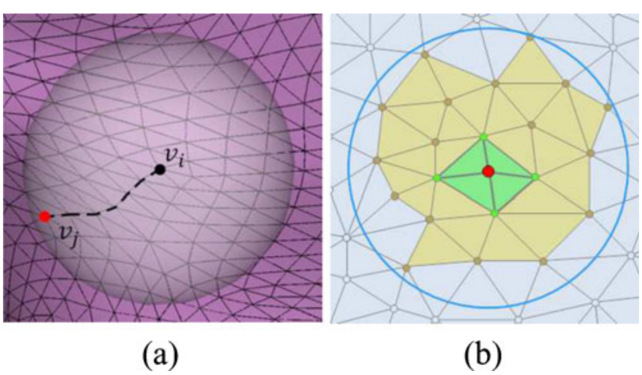



**Fig. 6** **a** Extended neighborhood of vertex $v_i$ and schematic geodesic distance to neighbor $v_j$. **b** First and extended neighborhoods around given vertex

structure is based on the fast marching method [21] adapted to our approach. Map is a data structure, implemented through the balanced tree, where each value is located according to its particular key. The extended neighborhood structure is built such that each vertex has a geodesic distance from the given central vertex at different scales of the integral volume descriptor. The algorithm is very efficient and has the complexity of $O(\log n)$ for every vertex, where $n$ is the number of neighbors in the extended neighborhood.

### 3.3 Integral volume descriptor computation

An integral volume descriptor is a discrete multiscale parameter-based curvature on local neighborhoods that makes it possible to deal with noisy data and to obtain invariants on multiple scales without any change of the given geometry. The local neighborhoods are defined via balls or spheres, whose radius defines the scale. A descriptor is a vector that is assigned to each vertex of both input models and computed as follows:

1. For a given vertex $v_i$ estimate the mean curvature $\widetilde{H}_k(v_i)$ over an extended neighborhood at each scale $k$ according to Section 2.2.
2. Substitute the mean curvature $\widetilde{H}_k$ and integration radius $r_k$ with respect to the scale into equation:

$$V_k(v_i) = \frac{1}{2} - \frac{3}{16}\widetilde{H}_k(v_i) \cdot r_k \tag{11}$$

Repeat steps 1–2 for each vertex of both input models.

The scale $k$ ($k$=6 in our implementation) over which the descriptor is computed, and the maximal radius $r_{\max}$ of the integration kernel are controlled by the user. The radius is sampled at discrete intervals by dividing the maximal radius $r_{\max}$ into $k$. The maximal radius $r_{\max}$ is usually set to 0.1 $L_{\max}$, where $L_{\max}$ is a maximal diagonal length of the model.

### 3.4 Feature selection

Feature selection focuses on picking a small set of points on the scanned data to simplify further identification of corresponding points on the CAD model. The selection process is based on analyzing the volume descriptors computed for each point of the scanned shape. The descriptor can be of any dimension. We use this multiscale property for finding and selecting the unique data points with uncommon descriptor values. A point is selected as a potential feature if its descriptor value is rare among all descriptors computed for the data set. The rare values are defined by the threshold. Most shapes contain these rare values at different scales, so we do not expect a point to be a feature over the entire scale space of the descriptor. From the set of potential features, we select as a feature a point whose descriptor value is rare over a set of at least $k/3$ consecutive kernel radii of the volume descriptor, where $k$ is a scale of the descriptor. In this case, the outliers may resemble features for some radii but are not persistent.

Let $V_k(p_i)$ be the integral volume descriptor assigned at each shape point with some value at each scale $k$. Here, we present an algorithm based only on one dimension of the descriptor. Feature selection for any other dimension can be implemented similarly. The feature selection proceeds as follows:

1. Compute a histogram of descriptor values, $f(p_i)$ for all points in $P$. The bins width $h_n$ in the histogram is computed using Scott's rule:

$$h_n = 3.49\,\sigma\,n^{-\frac{1}{3}} \tag{12}$$

where $\sigma$ is the standard deviation of the $n$th descriptor values [24].

The $\sigma$ is calculated by the following formula:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{13}$$

2. To select features, the least populated bins of the histogram is identified, such that the total number of points in these bins is smaller than a given maximum threshold $s$. In our implementation $s$=0:0.02$n$ ($n$ is a number of points in the model).
3. Since nearby points are likely to belong to the same feature, we want to prevent the algorithm from picking points that are too close to each other. Enforcing the minimal separation distance between the feature points also results in more stable configurations in the correspondence search stage of the algorithm.

Threshold $s$ controls the number of selected features and ensures their sufficiency. A relatively small number of selected features enable fast and efficient matching. Figure 6 illustrates the feature points picked on the blade model by using the proposed feature selection algorithm.

### 3.5 Correspondence search

After features were selected, we search for optimal correspondence to the points in the CAD model. The search process includes two main steps: (1) identification of a potential corresponding set and (2) verification for an optimal corresponding set. The identification of corresponding points is based on matching the descriptors values over all consecutive scales between both input shapes. As a result, for each feature point of the scanned data, we obtain a set of potential corresponding points on the CAD model. Given is a set of $n$ feature points picked on the scanned inspected model. Figure 7 illustrates the selected feature on the scanned inspected model and its potential corresponding set on the CAD model.

Since the number of the resulting potential corresponding points is still too large and not all of them are correct, we apply a verification algorithm to obtain the optimal correspondence set. The main idea of the algorithm is to identify equal triangles formed by triplets of points from each data set. As a result, the number of corresponding points at each feature point is significantly reduced. However, this set is not final and will still include incompatible corresponding points caused by symmetric identical shapes. The final verification of the corresponding sets is accomplished in the global registration stage. The overall identification time is dependent on the number of feature and their corresponding points in the data sets.

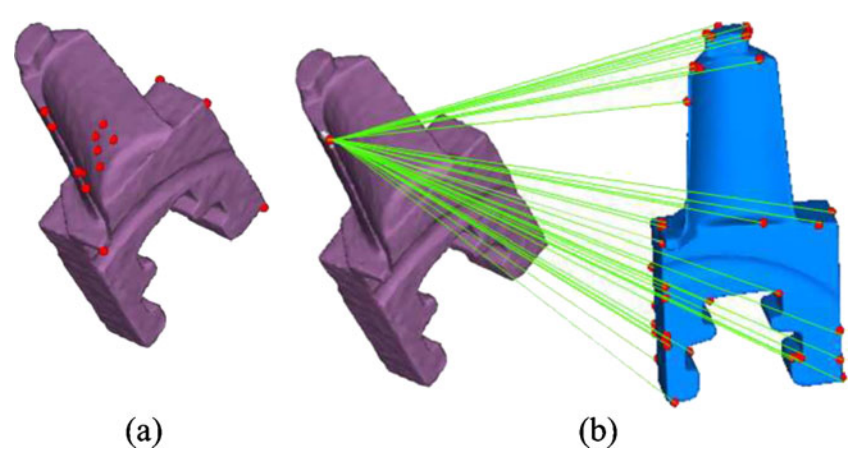



**Fig. 7** **a** Selected feature points. **b** Potential corresponding features

### 3.6 Registration

Once the optimal correspondence set has been built, the arbitrary positioned input shapes can be aligned globally and locally.

#### 3.6.1 Global registration

Starting from arbitrary initial positions of the inspected shape, we first look for a possible coarse alignment. Two 3D shapes can be aligned by mapping three noncollinear features of the inspected shape on their corresponding points from the CAD model. However, different triplets of features may yield various transformations. Therefore, incompatible corresponding points should be detected. For each triplet of features of the scanned model, the following algorithm is applied:

1. Calculate a transformation that maps a given triplet of features onto each triplet of their corresponding points from the CAD model.
2. For each calculated transformation, find the alignment error according to the following equation:

$$e = \frac{1}{nk} \sum_{i=1}^{n} \sum_{j=1}^{k} d^2\left(Rp_i + t, q_j^{s_i}\right) \tag{14}$$

where:

- $n$ is the total number of selected features
- $p_i$ is the current selected feature
- $k$ is the total number of corresponding points to a given feature in the corresponding set
- $q_j^{s_i}$ is the current point from the corresponding set

3. Select the transformation that has minimal alignment error and keep its corresponding point triplet. Remove the other corresponding points from the set.
4. Repeat steps 1–3 for all other triplets of features.
5. Select the best transformation over all triplets of features that bring two shapes close and have minimal alignment error.

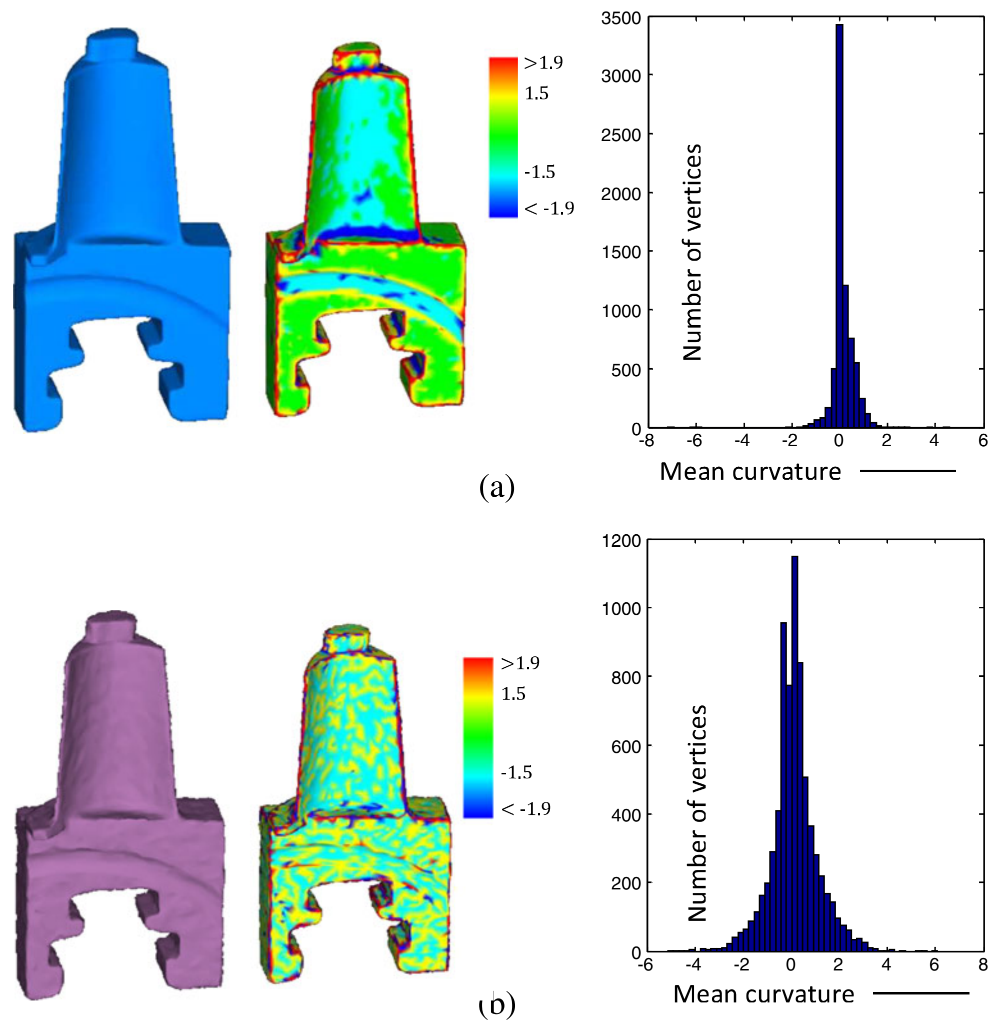



**Fig. 8** Mean curvature map on a synthetic mechanical part using angle-based method with mean curvature histogram. **a** The CAD model. **b** Scanned noisy part

As a result of this stage, we obtain optimal transformation parameters and the final corresponding set on the CAD model.

### 3.6.2 Local registration

After the coarse alignment, a small number of iterations of standard ICP with point-to-point error metric are applied, bringing the shapes into exact alignment. ICP transformation $T=(R,t)$ is initialized relative to the result of the global registration stage and using the final correspondence data sets. The following steps are iterated until the change in pose becomes very small:

1. Using the correspondence data, compute the transformation $T$ that minimizes the mean square error of Eq. 5.
2. Apply this transformation on the inspected shape.

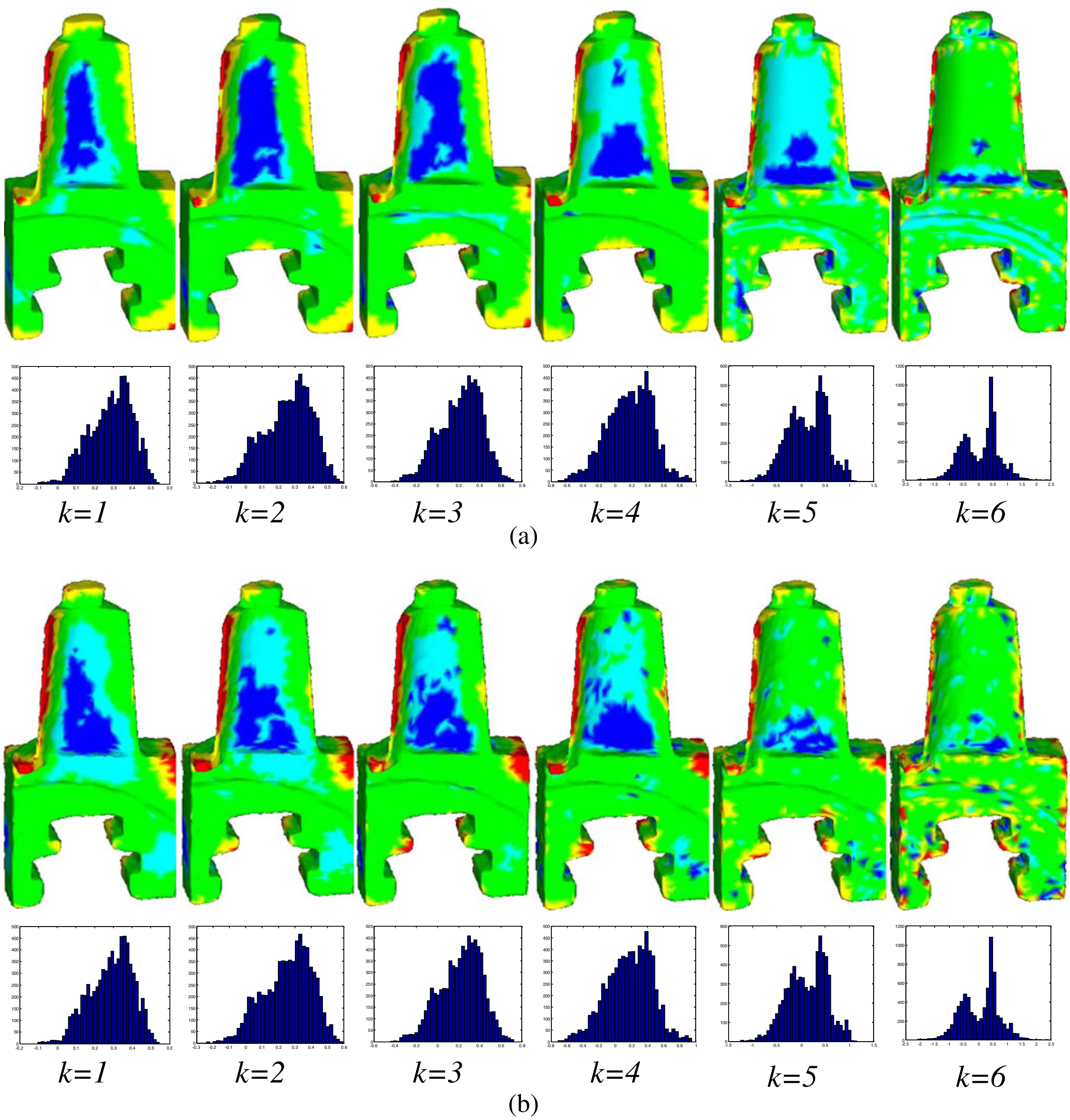



**Fig. 9** The integral volume descriptor map and the histogram of the (**a**) designed part and (**b**) scanned data

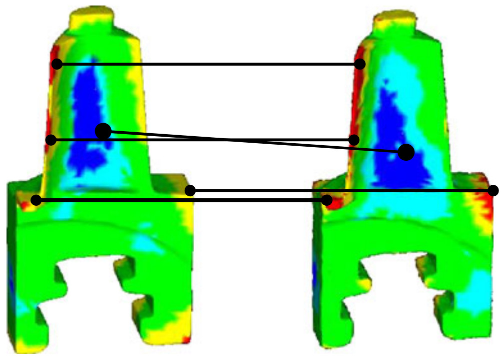


**Fig. 10** Similarity of several unique regions between the CAD model (*left*) and the scanned data (*right*) at scale $k=1$

3. Repeat steps 1–2 until the change in mean square error falls below the desired registration precision:

$$e_h + e_{h+1} < \varepsilon \tag{15}$$

where $\varepsilon$ is the threshold, and $h$ is the number of the current iteration.

In the verification process, volume descriptors of corresponding shapes are verified. As a result, the noncongruent regions between the two parts are emphasized, and the defects can be detected visually.

## 4 Examples and performance analysis

This chapter discusses the results and performance of several examples at several stages of the algorithm. Our algorithm has been tested on noisy synthetic models. The assumption underlying this work is that scanning noise can be represented by a Gaussian noise with a deviation of a scanned surface point along the normal direction. The original synthetic models were corrupted with additive Gaussian noise with zero mean: $\phi \sim N(0, \sigma^2)$, where $\sigma^2$ is the variance. All tests were performed on a PC Pentium 4, with 2.8 GHz and 512 MB of memory. The following scheme illustrates the model processing at each stage of the proposed algorithm.

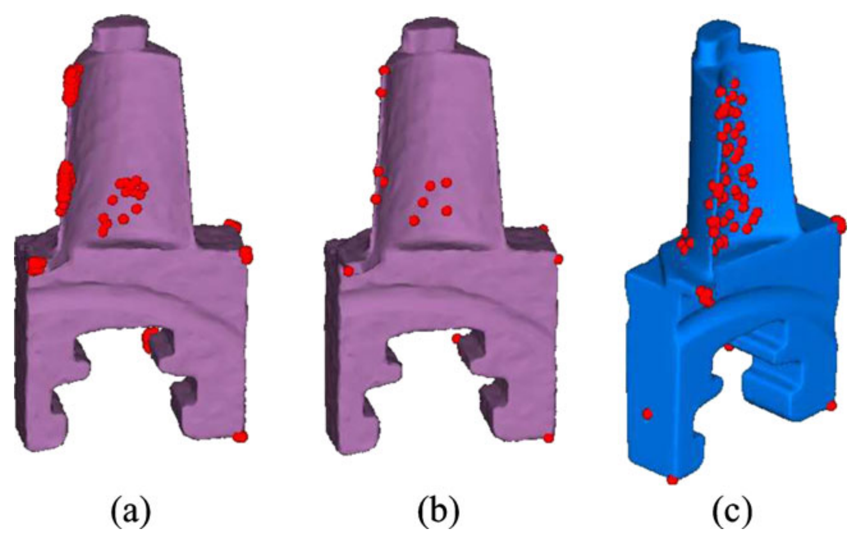
### 4.1 Applying angle-based method

In this section, the results of the angle-based method are provided for mean curvature estimation. One model represents a CAD model (Fig. 8a), and the other model represents the scanned detail with Gaussian noise of ±2% added along the normal direction (Fig. 8b). As seen in the above figure, noise significantly affects the value of the mean curvature. The mean curvature of the original part is uniformly distributed, as indicated by the homogeneous colors and the narrow histogram. On the other hand, for the scanned part the mean curvature is nonuniformly distributed, as shown in the nonhomogeneous colors and the wide histogram. The main conclusion from this section is that registration cannot be based on mean curvature alone.

### 4.2 Applying feature mapping

In this stage, the integral volume descriptor is computed for each vertex at six different scales $(k=1{:}6)$. The resulting descriptor map is presented in Fig. 9, where the unique shapes are colored according to frequency of their descriptor values at each scale. The red/blue regions represent the most unique shapes of the model. The common shapes, whose vertices belong to the bins of the histogram with a number of points ranging from $0.02n$ to $0.04n$, are colored in yellow/pale blue according to the sign of the descriptor.

**Fig. 11** Features selected on the scanned data with (**a**) $R_s=0$ and (**b**) $R_s=0.03r_{max}$. **c** The corresponding points on the CAD model



The scales in Fig. 9 increase from left to right, i.e., the map in the left column presents the smallest descriptor scale, and the map in the right column presents the biggest descriptor scale.

As seen from the above figure, the advantage of the integral volume descriptor is its robustness to noise. Persistent unique regions appear over different scales of each model. Based on these persistent unique regions, we can select features. Furthermore, we can easily identify the similarity in the detected unique regions of the input models at the same scale, despite the presence of noise. Figure 10 illustrates the similarity between several unique regions in the input models at scale $k=1$.

### 4.3 Applying matching

The matching stage includes three main steps: feature selection on the scanned data; finding potential corresponding points on the CAD model; finding optimal corresponding points on the CAD model. All vertices that have unique descriptor values over at least five scales are selected (see Fig. 11a). For each feature, one representative point is selected. For a given vertex, all neighbors that fall into a circle of radius $R_s$ around the given vertex are marked and cannot be selected (see Fig. 11b). This leads to more stable results in the correspondence search stage.

After selecting features on the scanned data, we identify their corresponding points on the CAD model. Matching is based on comparing integral volume descriptor values over a set of at least five scales (Fig. 12). The matching data between separated pairs of some correspondences are presented in Fig. 13. As seen in the figure, there is significant error in the mean curvature between the correspondences (marked near the selected vertex). Such error in mean curvature results from the presence of significant noise. However, the error in the integral volume descriptor (see the graph) between the input shapes is not more than 2% except in rare instances, despite the noise. Since we want to ensure that true correspondences are found, a comparison threshold is determined ($s$=2%). As a

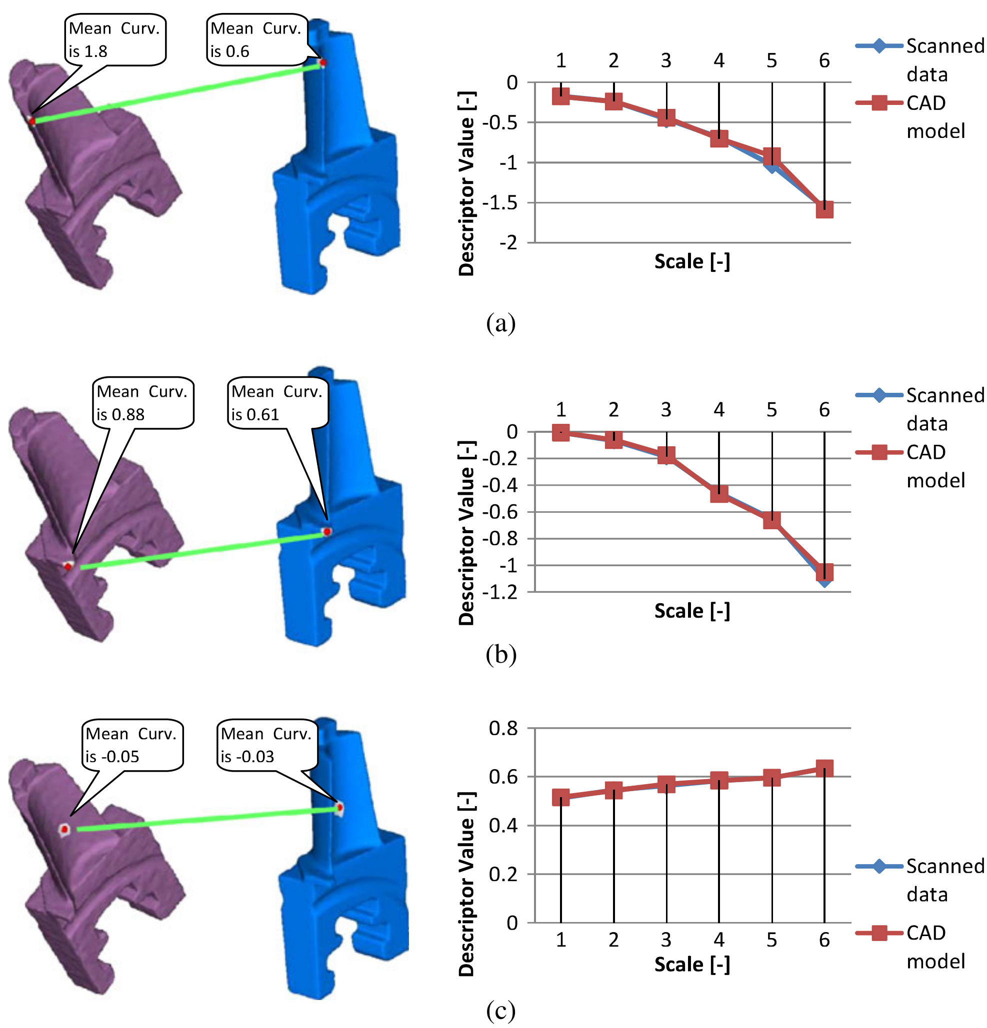



**Fig. 12** Matching data between free correspondences

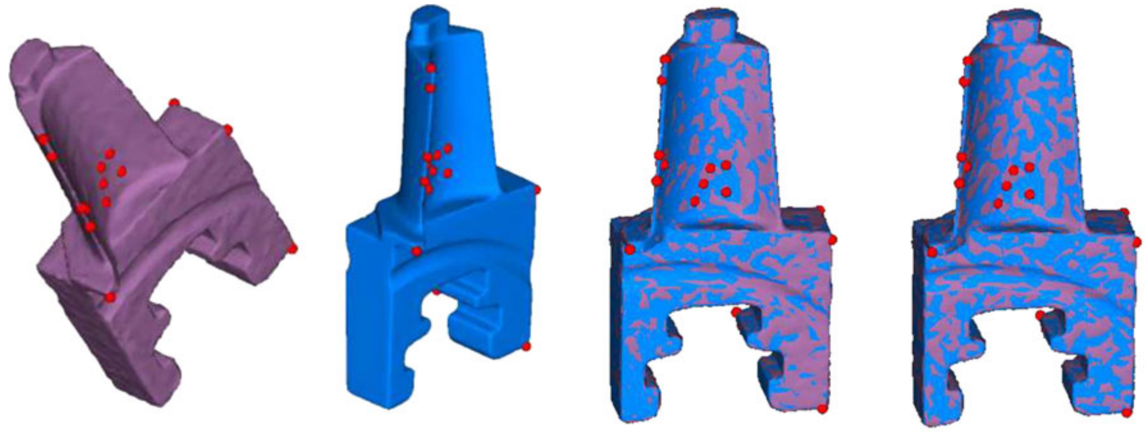



**Fig. 13** **a** Features selected on the scanned data. **b** Potential corresponding features on the CAD model. **c** The global registration. **d** The local alignment

result, for each feature on the scanned data a set of potential corresponding points is obtained on the CAD model. Then, the set of optimal corresponding points are detected, and false feature points are eliminated (Fig. 13b).

### 4.4 Applying registration and alignment

First, we apply global registration by finding the best transformation between triplets of correspondences. The global registration brings two input models into coarse alignment (Fig. 13c). After the coarse alignment, the standard ICP with point-to-point error metric is applied iteratively, bringing the models into fine alignment (Fig. 13d). ICP is initialized relative to the global registration.

### 4.5 Performance analysis

We have applied our algorithm pipeline to a CAD model of a blade. We align the model to a copy of itself which has been corrupted by zero mean Gaussian noise. The input models were initialized at an arbitrary orientation and position. The input models size—7,241 vertices, the running time—24.1 s, the number of features selected on scanned data—18, the number of potential corresponding points detected on the scanned model—214, the number of optimal corresponding points detected on the CAD model—18, the global registration error—1.3 (distance$^2$), and local registration error—0.019 distance$^2$. Table 1 summarizes the time complexity for phases of the presented algorithm. The size parameters are: $n$—model size (number of vertices); $e$—maximal number of vertices in an extended neighborhood; $f$—number of features selected on the scanned data; $c_p$—size of the potential feature corresponding set; $i$—number of iterations to achieve desired convergence. Summing it all up, we conclude that the time complexity of the overall algorithm is bounded from above by $O(n \cdot e \cdot \log e)$.

## 5 Conclusions

This work has proposed a method for automatic feature-based registration and alignment of two 3D freeform shapes, one from the scanned data and the other from the CAD model. The scanned manufactured part usually includes defects and distortions. The method makes no assumptions about their initial positions. The proposed algorithm uses a multiscale shape descriptor to select features on the scanned data and identify their corresponding features on the CAD model. The proposed shape descriptor depends on the mean curvature, is invariant with respect to local shapes, and is robust to noise. A coarse alignment is computed by finding and registering the best matching triplet of features. This resulting coarse alignment is used by the ICP algorithm to achieve a tuned alignment. The proposed method is

**Table 1** The time complexity of the proposed algorithm

| Phase | Time |
|---|---|
| Angle-based method | $O(n)$ |
| Integral volume descriptor computation | $O(n \cdot e \cdot \log e)$ |
| Feature selection | $O(n)$ |
| Potential corresponding set identification | $O(n \cdot f)$ |
| Optimal corresponding set identification | $O(f^3 \cdot c_p)$ |
| Global registration | $O(f^4)$ |
| Local registration | $O(f \cdot i)$ |
| Total | $O(n \cdot e \cdot log\, e)$ |

automatic, efficient, and straightforward to implement. The algorithm can also be effective in the case of partial scanned inspected shapes. The feasibility of the proposed method is demonstrated on a number of freeform engineering and medical models. The proposed method has the following limitations: An important factor affecting the performance of our selection method is the density of the sample. For denser scanning, the features are extracted more accurately. The algorithm works partially on weak point-like features of the scanned model.

## References


1. Spyridi AJ, Requicha AG (1990) Accessibility analysis for the automatic inspection of mechanical parts by coordinate measuring machines. IEEE international conference on robotics and automation, Cincinnati
2. Farin G (1988) Curves and surfaces for CAGD. A practical guide. Academic, San Diego
3. Hameiri Y, Shimshoni I (2002) Estimating the principal curvatures and the Darboux frame from real 3D range data. At 1st International Symposium on 3D Data Processing Visualization and Transmission (3DPVT 2002)
4. Stokely EM, Wu SY (1992) Surface parameterization and curvature measurement of arbitrary 3-D objects: five practical methods. IEEE Trans Pattern Anal Mach Intell 14(8):833–840
5. Lipshitz B, Fischer A (2005) Verification of scanned engineering parts with CAD models based on discrete curvature estimation. Journal of Computing and Information Science in Science and Engineering 5(2):116–117
6. Meek DS, Walton DJ (2000) On surface normal and Gaussian curvature approximations given data sampled from a smooth surface. Comput Aided Geom Des 17:521–543
7. Wolfson H, Rigoutsos I (1997) Geometric hashing: an overview. IEEE Computer Science and Engineering 10–21
8. Hamann B (1993) Curvature approximation for triangulated surfaces. Computing Suppl 8:139–153
9. Taubin G (1995) Estimating the tensor of curvature of a surface from polyhedral approximation. Proceedings Intl Conf on Computer Vision 902–907
10. Do Carmo MP (1976) Differential geometry of curves and surfaces. Prentice-Hall, Englewood Cliffs
11. Meyer M, Desbrun M, Schröder P, Barr AH (2002) Discrete differential geometry operators for triangulated 2-manifolds. Proceedings of VisMath '02 Berlin, Germany
12. Gelfand N, Mitra N, Guibas L, Pottmann H (2005) Robust global registration. Proc Eurographics Symp Geom Processing 197–206
13. Stockman G (1987) Object recognition and localization via pose clustering. CVGI 361–387
14. Eggert DW, Lorusso A, Fisher RB (1997) Estimating 3D rigid body transformations: a comparison of four major algorithms. Mach Vis Appl 272–290
15. Besl PJ, McKay ND (1992) A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell 14(2):239–256
16. Sharp GC, Lee SW, Wehe DK (2002) ICP registration using invariant features. PAMI 24(1):90–102
17. Barequet G, Sharir M (1997) Partial surface and volume matching in three dimensions. PAMI 929–948
18. Manay S, Hong B,Yezzi A, Soatto S(2004) Integral invariant signatures. In ECCV 87–99
19. Fishkel F, Fischer A, Ar S (2006) Verification of engineering models based on bipartite graph matching for inspection applications, Lecture Notes in Computer Science, Geometric Modeling and Processing—GMP. Springer, pp 485–499
20. Kase K, Makinouchi A, Nakagawa T, Suzuki H, Kimura F (1999) Shape error evaluation method of free-form surfaces. Comput Aided Des 31(8):495–505
21. Kimmel R, Sethian JA (1998) Computing geodesic paths on manifolds. In Proceedings of the National Academy of Sciences No. 95, 8431–8435
22. Pottmann H, Huang Q-X, Yang YL, Kolpl S (2005) Integral invariants for robust geometry processing. Technical Report 146. Vienna University of Technology
23. Fischer A (2000) Multi-level models for reverse engineering and rapid prototyping in remote CAD systems. Comput Aided Des 32 (1):27–38
24. Scott D (1979) On optimal and data-based histograms. Biometrika 66:605–610