# Free-form Surface Matching for Surface Inspection

**Edvaldo M. Bispo and Robert B. Fisher**

*Dept. of Artificial Intelligence, University of Edinburgh*

## 1 Introduction

Despite their great accuracy, coordinate measuring machines (CMM) for the inspection of mechanical parts have some drawbacks related to their low speed, which make it impossible to measure many points on the surface of the objects. Another difficulty is in programming the CMMs which make their use very cumbersome. We are investigating the use of dense range data acquired using laser stripers for the inspection of mechanical parts. Most laser stripers do not produce as accurate measurements as CMMs but are much faster and consequently are able to measure many more points on the surface of the object. This makes possible a better evaluation of shape. We concentrate on the inspection of free-form surfaces, which are particularly interesting because few inspection processes can inspect the surface shape across the whole surface.

The main idea behind our research is the comparison between the three dimensional geometric information present in the range image and descriptions or models of ideal parts stored previously by the system.

Given a part to be inspected and a corresponding model of the part stored in the *model data base*, the first step towards inspecting the part is the *acquisition of data* corresponding to the part which in our case means the acquisition of a range image of it. Before any comparison between data and model can take place, it is necessary to align the model with the range data of the part. We use a modified version of Besl's ICP (iterative closest point) algorithm [10] that uses *a priori* knowledge of an approximation of the right registration to make the method more robust to outliers and avoid the problem of convergence to local minima.

After registration, we can explicitly compare the complete measured surface to the model and inspect the part. Because the manufacturing and measurement processes are never perfect, it is necessary to take account of the part's tolerances, *i.e.* the dimensions of the part should be within the tolerance values explicitly determined in the model.

In this paper we start by a brief description of the kind of shape model we used in our experiments and how we acquired these models in Section

**3**. In Section **4** we comment on the registration of our spline models with range data. The inspection of the surface's profile after registration is discussed on Section **5**. The detection of periodic deformation on the surfaces is discussed in Section **6**. Final comments and conclusions are presented in Section **7**.

## 2 Model shapes

Because we were particular interested in the inspection of sculptured surfaces (*e.g.* turbine blades), we decided to use B-splines for modelling the nominal shape of the objects being inspected. In all the experiments described in this paper we used as model shape uniform B-splines. The models used here were obtained by fitting B-splines to range data; however, in real applications the surface spline descriptions would be obtained from a CAD model of the part.

Figures (3) and (4) show the spline models built for the test objects **1** and **2** shown in Figures (1) and (2). The average distance between model and data points in both cases was on the order of tenths of millimeters (with image noise standard deviation equal to 0.15 mm). As one would expect the bigger errors happened near the discontinuities of the image. This fact explains the better results obtained with the test object number **2** which has fewer discontinuities than the test object number **1**.

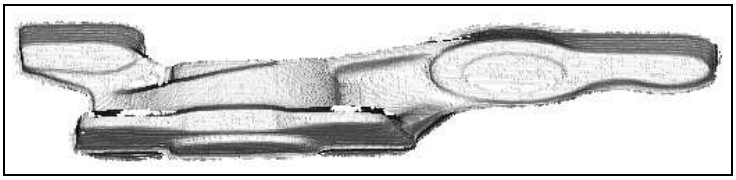

**Figure 1.** Range image of Test object number 1

## 3 Registering free-form surfaces

Given a range image of a free-form surface, which describes the shape of this surface in the sensor coordinate system, and a model of the same surface in a different coordinate system, the problem of registering model and data consists of finding the rigid transformation (rotation and translation) that aligns or superposes both.

Some of the first research into matching free-form surfaces was done by Faugeras and Hebert [6]. They presented a matching algorithm able

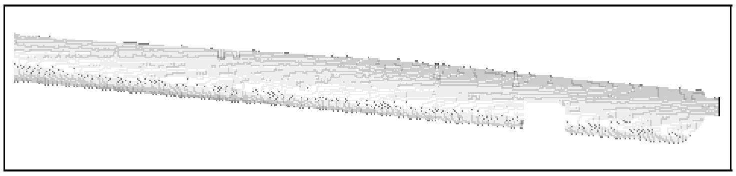

**Figure 2.** Range image of Test object number 2

to successfully match a Renault auto part, but their algorithm had the drawback of depending on the existence of planar regions in the object being matched. Potmesil [14] developed a system for modelling the complete surface of a object from different range images. The different range images were matched through heuristic search in the 6-D space of the translation and rotation. Chen and Medioni [11] also developed an algorithm for registering overlapping range images of an object. They assumed an *a priori* knowledge of an estimated pose aligning the different range images and approximate one of the surfaces being matched using its tangent planes. Besl and McKay proposed the ICP (Iterative Closest Point algorithm) [10] for registering free-form surfaces using only the 3-D points on the surface of the object. Zhang [18] proposed an algorithm very similar to the ICP in which an *a priori* approximation of the registration is assumed to be known and is used to accelerate the convergence of the algorithm and make it more robust to outliers. Recently Stein [17] advocated the use geometric hashing together with an IT (Interpretation Tree) [9] for recognizing and aligning objects in range and gray level images.

We assume that the object being registered does not present any salient features (*e.g.* planar regions) that could help guide the registration process and concentrate solely on the use of the 3-D points for obtaining the registration. We also assume the *a priori* knowledge of an approximation of the pose aligning model and data and investigate the use of algorithms such as the ICP for improving the accuracy of the registration.

The final accuracy that can be obtained using only 3-D points on the data and model surface depends critically on the calculation of the pose aligning sets of corresponding 3-D points on the surfaces of model and data. Therefore we start by evaluating the accuracies that can be achieved using sets of corresponding 3-D points in Section 3.1. In Section 3.2 we discuss the ICP algorithm in detail and present some extensions for improving the robustness of the algorithm.

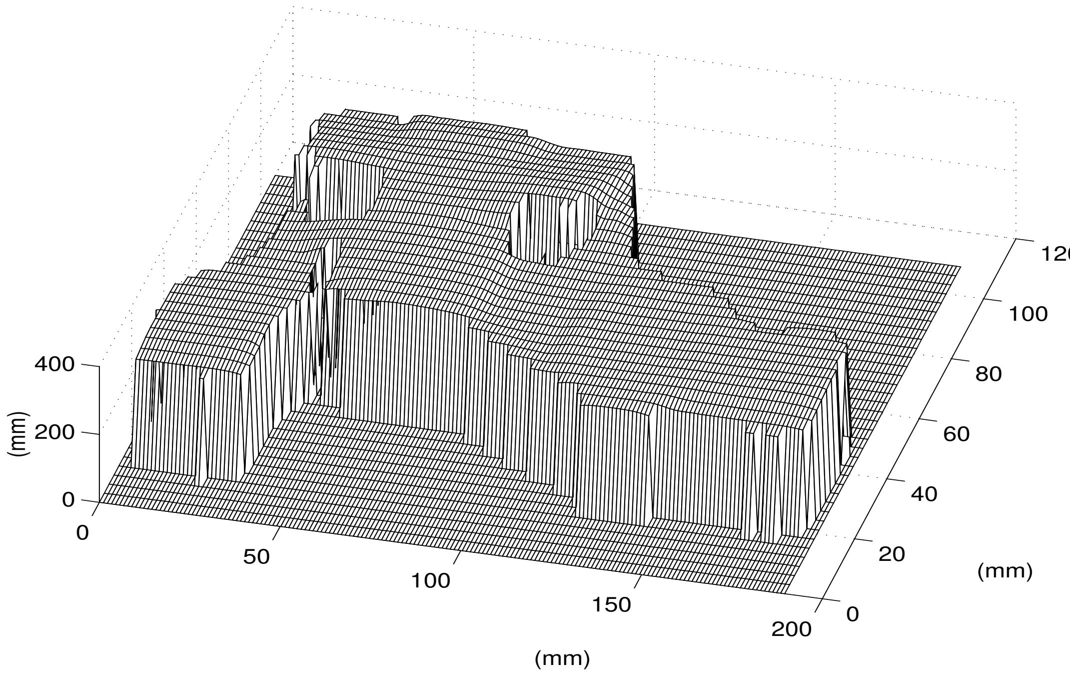


**Figure 3.** Uniform B-spline model of Test object number 1. Average distance between model and data is 0.4 mm. Uniform grid of 50X50 knots.

### 3.1 Registering two sets of corresponding 3-D points

Consider a set of data points $\mathbf{D} = \{d_i\}_{i=1}^{N}$ and a set of corresponding model points $\mathbf{M} = \{m_i\}_{i=1}^{N}$ such that $\mathbf{D}$ and $\mathbf{M}$ are related by a rigid transformation corresponding to a translation $\mathbf{T}$ and a rotation $\mathbf{R}$ as defined in equation (3.1).

$$d_i = \mathbf{R}m_i + \mathbf{T} + w_i \tag{3.1}$$
$$\mathbf{E}[w_i] = [0 \;\; 0 \;\; 0]^t \tag{3.2}$$
$$\mathbf{E}[w_i \cdot {w_i}^t] = \begin{bmatrix} {\delta_i}^2 & 0 & 0 \\ 0 & {\delta_i}^2 & 0 \\ 0 & 0 & {\delta_i}^2 \end{bmatrix} \tag{3.3}$$

where $w_i$ is white noise modelling the sensor errors and $\mathbf{E}[\cdot]$ is the expectation operator.

The problem of finding $\mathbf{T}$ and $\mathbf{R}$ that minimizes the mean square distance $E_M$ between $\mathbf{D}$ and $\mathbf{M}$ (equation (3.4)) is a well known problem to which there are several proposed solutions [4, 5, 6]. Under the the criterion of accuracy in finding the right values of $\mathbf{R}$ and $\mathbf{T}$ all these methods are

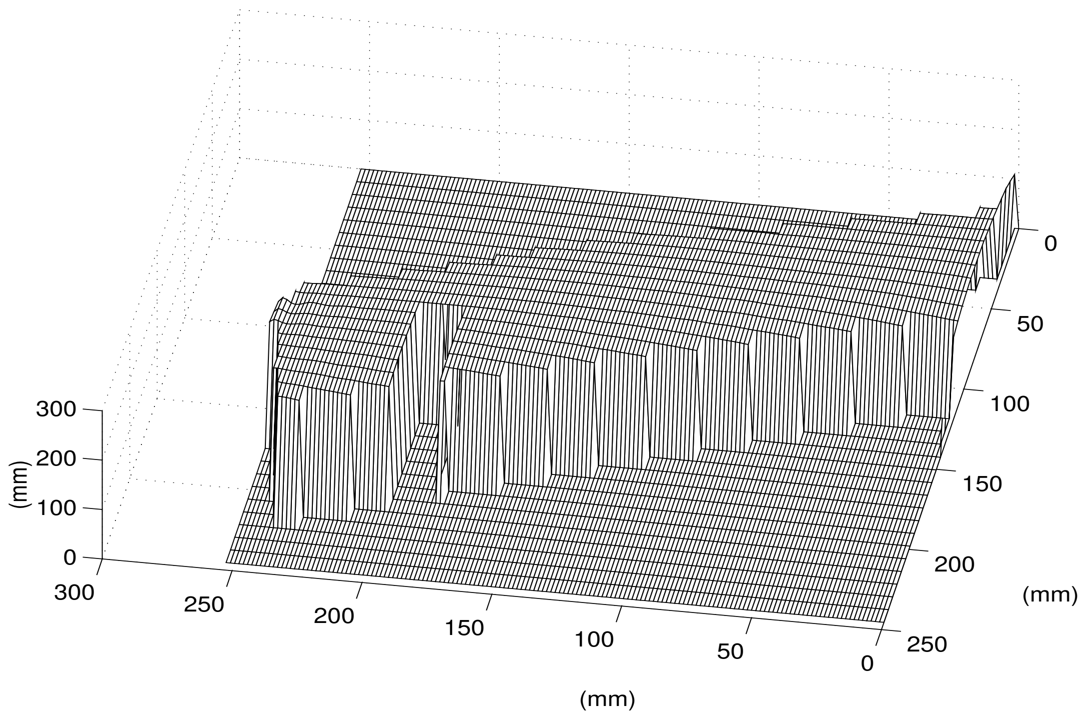


**Figure 4.** Uniform B-spline model of Test object number 2. Average distance between model and data is 0.08 mm. Uniform grid of 50X50 knots.

equivalent, and we will concentrate our attention on Horn's algorithm [5].

$$E_M \quad = \quad {D_M}^2 = \sum_{i=1}^{N} \|d_i - \mathbf{R}m_i - \mathbf{T}\|^2 \tag{3.4}$$

In order to investigate the degradation in the accuracy of the algorithm with increasing noise $\delta_i$ we extracted 3-D points from the spline models of test objects **1** and **2** and use them as model points. The data points were generated by applying a known rotation and translation to the model points and corrupting the result with noise. The experiment was run for different values of rotation and translation and different levels of noise $\delta_i$. To each value of noise level 200 samples were considered.

As one could expect the results did not depend on the specific values of rotation and translation. Also, the final value of the mean square distance ($E_M$) found by Horn's algorithm tended asymptotically to $\mathbf{E}[{w_i}^t \cdot w_i] = 3 \cdot \delta^2$ as the number of matched points increase. Figures (5) and (6) show the variation of $D_M$ (eq. 3.4) and the error of the translation estimate ($\hat{\mathbf{T}}$) for the test object number **1**. These graphs show the maximum accuracy that can be achieved for a given level of noise, which corresponds to the combined effects of sensor and modelling errors, and number of matched points. The graphs also show that the error in the estimate of translation will decrease almost inversely with the number of matched points, as shown in equation (3.5). It is important to remember that the graphs show the very best accuracy that can be achieved, because other sources of errors such as sampling errors and mismatchings were not taken into account. The experiments with test object **2** produced very similar results.

$$\|\hat{\mathbf{T}} - \mathbf{T}\| \quad \propto \quad \delta_i / N \tag{3.5}$$

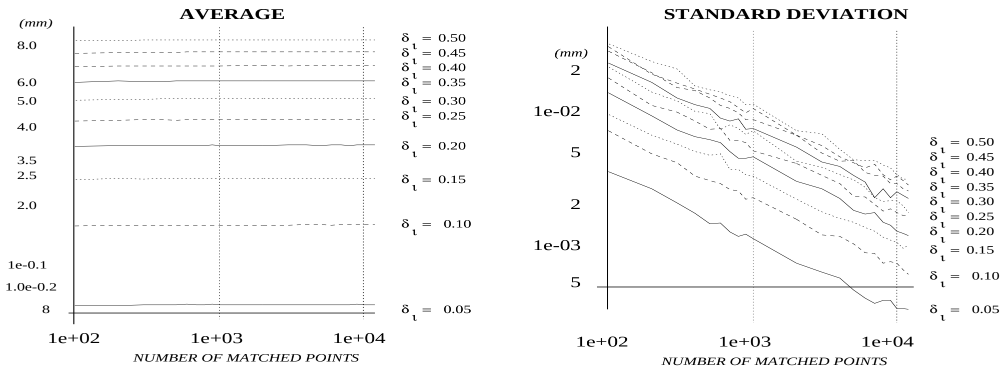


**Figure 5.** Variation of the mean distance ($D_M$) with the noise level $\delta_i$ for test object number **1**. Average and standard deviations calculated using 200 samples (10 different curves for $\delta_i$ varying from 0.05 mm to 0.5 mm).

### 3.2 The ICP algorithm

Given a set of data points $\mathbf{D} = \{d_i\}_{i=1}^{N}$, a model shape $\mathbf{M}$ and an initial estimate $(\hat{\mathbf{R}}^{\mathbf{0}}, \hat{\mathbf{T}}^{\mathbf{0}})$ of the rigid transformation aligning $\mathbf{D}$ and $\mathbf{M}$, the ICP algorithm [10] is:

1. Apply the rotation $\hat{\mathbf{R}}^{(\mathbf{k})}$ and translation $\hat{\mathbf{T}}^{(\mathbf{k})}$ to the points in $\mathbf{D}$ obtaining the set of points $\mathbf{D}^{(\mathbf{k})} = \{d^{(k)}{}_i\}_{i=1,N}$
2. To each point $d^{(k)}{}_i$ find the point $m^{(k)}{}_i$ in $\mathbf{M}$ which is closest to $d^{(k)}{}_i$.
3. Calculate new estimates of $\hat{\mathbf{T}}^{\mathbf{k+1}}$ and $\hat{\mathbf{R}}^{\mathbf{k+1}}$ by assuming the points $d_i$ and $m^{(k)}{}_i$ correspond, and applying Horn's algorithm (or equivalent).

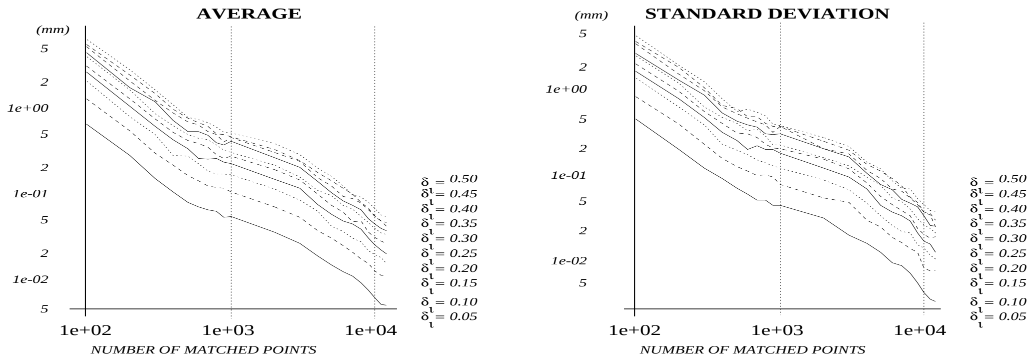


**Figure 6.** Variation of the translation error ($\|\hat{T} - T\|$) with the noise level $\delta_i$ for test object number **1**. Average and standard deviations calculated using 200 samples (10 different curves for $\delta_i$ varying from 0.05 mm to 0.5 mm).

4. Repeat **1** to **3** till convergence ($E_M$ stops decreasing).

where the superscript **k** corresponds to the iteration number.

In our experiments the model shape **M** was represented by a B-spline model of the surface being registered. Because the evaluation online of the points on the spline would be too computationally expensive, we used as the model shape **M** a dense grid of points extracted offline from the spline model.

A crucial step in the algorithm consists of finding the closest model point to the point ${d^{(k)}}_i$. Zhang [18] used K-D trees. We used a multiscale search for finding the closest point. This approach can occasionally fail when the region containing the real closest point is not detected during the coarse scale search. In our experiments this never happened because we start the search at a scale small enough to avoid this problem. Also, due to the consistency checks we added to the ICP algorithm (see below), occasional failures to find the true closest point do not degrade the performance of the algorithm. Figure 6 shows that an accuracy of the order of a tenth of millimeter (*i.e.* approaching the noise variance) can be achieved using only 1000 points. Therefore, we could subsample the data points (range image) which reduces considerably the number of points to be considered. The sampling of the model and data shape together with the use of the multiscale shape allowed us to achieve a good speed without the use of K-D trees. As suggested by Figure 6, better accuracies can be achieved by running an extra iteration of the ICP algorithm with more data points, after its convergence using coarser grids of data points.

In our experiments with the ICP algorithm, we verified most of the properties described by Besl [10]. Clearly, the biggest problem with the

ICP is the problem of converging to a local minimum of $E_M$ that does not correspond to the right registration between model and data. This problem may not be very significant with objects that have a large radius of convergence for the global minimum (*e.g* object test **1** to which the ICP will always converge to the right answer as long as the model was rotated less than 100 degrees around any axis), because it is only necessary to run the ICP a few times using different initial estimates of $\mathbf{R}$ and $\mathbf{T}$. But this fact is a very serious problem in the case of objects like the object test **2** that has a very small radius of convergence due to the existence of other local minima near the desired global minimum.

A interesting aspect of the problem of the initial estimate is the influence of the original translation estimate. As said in [10], when the data set is a subset of the model which covers a reasonable portion of the model, the ICP performance is reasonably insensitive to the original estimate of translation. However the final accuracy of the registration found by the algorithm can change considerably as illustrated in Figure 7 (a). The situation becomes even worse when the data covers only a small part of the model as seen in Figure 7 (b).

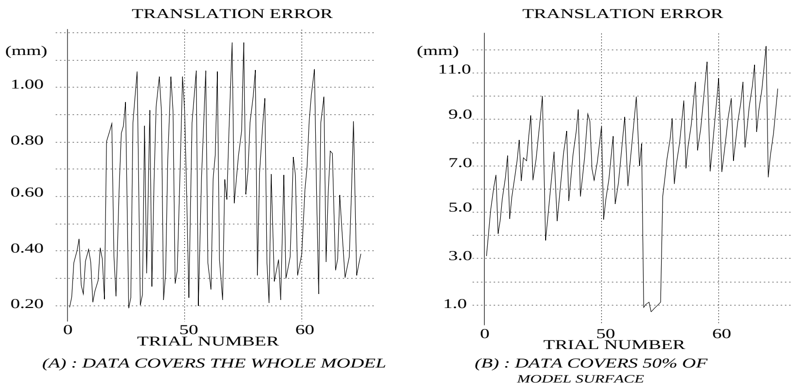


**Figure 7.** Variation of translation error ($\|\hat{\mathbf{T}} - \mathbf{T}\|$) of ICP with the value of the initial estimate of translation - Test object **1**. Initial estimates of translation vary between -2 and 2 times the dimensions of the object. Part (a) covers the complete model and part (b) uses data from a subset of the model.

Another problem with the ICP is the fact that it cannot cope with situations in which the data points are not a subset of the model points. This is a considerable limitation in applications such as model acquisition and inspection in which this assumption does not hold. In order to make the method more robust to outliers we altered the original algorithm and eliminated the pairs $(d_i, m^{(k)}{}_i)$ that did not satisfy a few consistency tests. The first of these tests, essentially used by Zhang [18] though in a different way, eliminates pairs of points $(d_i, m^{(k)}{}_i)$ if the distance between $d_i$ and

$m^{(k)}{}_i$ is bigger than three times the value of $E_M$ found so far. We also used two local geometric constraints for rejecting wrong pairs: the distance between points in the grid and the angles between the vectors joining pairs of points in the grid. Because of the surface sampling, a matching for a data point $d_i$ is any model point $m_i$ such that, after the alignment between model and data, $m_i$ belongs to the portion of the data surface $\mathbf{D}$ that $d_i$ represents. We approximated this region associated with $d_i$ by a cuboid centered on $d_i$ with the faces parallel to the planes $\mathbf{xy}$, $\mathbf{xz}$ and $\mathbf{yz}$ of the sensor coordinate system and dimensions corresponding to the size of the sampling grid.

The maximum values of distances and angles associated with the points $d_i$ were calculated considering the angles and distances between the cuboids associated with each of the data point. All the calculation of minima and maxima was done offline. In execution, whenever the relative distance between pairs of data points or relative angles between triples of data points was inconsistent with their corresponding model points, the corresponding pair of data points and model points were rejected.

These alterations make the method more robust to outliers but impose a couple drawbacks. All the tests proposed can only be used when the algorithm starts from a reasonable estimate of $\mathbf{R}$ and $\mathbf{T}$. Besides that, the consistency tests make the algorithm lose its monotonicity in the presence of outliers, *i.e.* the value of $E_M$ may increase sometimes. This happens because the tests may fail to eliminate all the outliers in some iterations. However, in our experiments with the modified ICP, the algorithm always ended up converging.

## 4 Surface profile measurement

Because manufacturing methods are incapable of producing parts with perfect shapes and sizes, an essential point in the design of the mechanical parts is the determination of the part tolerances. Several different approaches have been proposed for combining the dimensioning and tolerancing standards with solid modelling, see [12] for a detailed survey, but none of them were completely successful mainly because of the informal definitions given in the current standards. We intend to base our approach to the definition of tolerances on the theory of geometrical tolerancing proposed by Requicha [15, 16].

In essence, Requicha's theory says that :

- an actual feature $f_M$ is within tolerance $t$ if the measured feature $f_M$ lies inside a tolerance zone defined by offsetting the corresponding nominal feature $f_N$ by a positive distance $d_p$ and a negative distance $d_n$ such that:

$$t = d_p + |d_n| \tag{4.1}$$

- each feature is given tolerance values which define acceptable variations in *size, orientation* and *position.*

In this paper we use an adaptation of Requicha's theory, inspired on the work of Menq [7], to define position tolerance zones for measuring the profile of sculptured surfaces. As in the ISO standard, we define the profile tolerance zone as the region limited by two surfaces enveloping a sphere of diameter $t$.

The inspection of the surface profile consists then in:

**Registration :** Model and range data are registered using the ICP algorithm.

**Calculation of residual image :** Calculate the residual image $r(x, y)$ corresponding to the projection of the Euclidian distance between each point $(x, y)$ on the range image belonging to the surface being inspected and its closest point in the model on the direction of the normal of the model surface.

**Surface profile measurement :** Process residual image to verify if the surface is inside the tolerance zone.

Once the size of the tolerance zone $t$ is defined, the inspection process is very simple and consists in verifying if all the residual values are within a given threshold. The determination of the threshold value and its relation to the number of the points tested and the significance of the result is discussed in [7].

As an example, let us consider a case in which a random gaussian-distributed deformation of amplitude 0.1 mm is added to the center of the range image of the test object number **1**. Figure 8 shows the detected deformed region after comparing the residual of the registration with the threshold 0.05.

## 5 Detecting periodic deformations

In this section we will concentrate on the detection and measurement of waviness errors on sculptured surfaces. Waviness errors are caused fundamentally by the scalloped or cusped surfaces left by spherical and ball-nose cutting. This means that waviness errors are periodic deformations created by the cutting tools. Because periodic shape deformations produce peaks in the frequency space, we detect the periodic deformations by looking at the module of the discrete Fourier transform (DFT) of the difference image $\delta(x, y)$ obtained after the registration between model and data. The calculation of DFT is done in different subwindows covering the whole image and for each window the inspection process consists of:

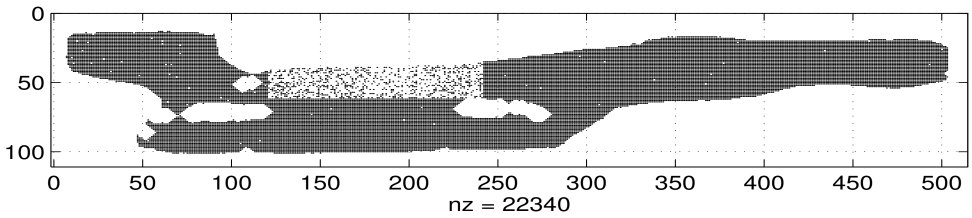


**Figure 8.** Region of range image out of position tolerance. The white pixels are the unmodeled background, the black pixels are the modeled part that is within tolerances and the textured grey pixels are the detected out-of-tolerance region.

**Deformation detection :** The DFT of the difference image in a window covering part of the original range image is thresholded. Peaks in the frequency domain bigger than a threshold value ($\tau_1$) indicate the presence of deformation.

**Localization of deformation region :** If a deformation is detected, the region of the deformation is found by applying the inverse DFT to the selected peaks of the DFT of the difference image, and then keeping the deformations whose absolute values are bigger than a threshold value ($\tau_2$).

A fundamental step in the inspection process is the definition of the threshold values ($\tau_1$ and $\tau_2$) and the size of the window in which the DFT is applied. These values depend on:

- the kind of deformation we want to detect,
- the minimum amplitude and size of deformation we want to be able to detect,
- the statistical model of the noise in the original range image,
- the statistical model of the random error of the manufacturing process,
- the maximum acceptable probability of false alarm ($P_f$),
- the maximum acceptable probability of not detecting a given deformation ($P_n$) and
- the registration errors.

It is important to notice that an important limitation in the detection ability is the spatial frequency of the deformation. In order to avoid aliasing problems, the spatial period of the deformation should be bigger than 2 pixels.

If we ignore the registration errors and assume that the random error in the manufacturing process is much smaller than the noise in the range image, the residual between model and data after registration can be written as in equations (5.1) to (5.6). In these equations $f(x,y)$ correspond to the periodic deformation we want to detect and $\eta(x,y)$ to the noise in the range image.

$$\begin{aligned}
\delta(x,y) &= f(x,y) + \eta(x,y) && (5.1)\\
\mathbf{E}[\eta(x,y)] &= 0 && (5.2)\\
\mathbf{E}[\eta(x,y)\cdot\eta(x,y)] &= \sigma^2 && (5.3)\\
\mathbf{E}[\eta(x,y)\cdot\eta(x',y')] &= 0 \quad (x \neq x' \ \ or \ \ y \neq y') && (5.4)\\
x &= \{0,1,...,N-1\} && (5.5)\\
y &= \{0,1,...,N-1\} && (5.6)
\end{aligned}$$

where : N is the size of the window considered.

The DFT of $\delta(x,y)$, that we will call $R(u,v)$, is given by:

$$\begin{aligned}
R(u,v) &= \frac{1}{N}\cdot\sum_{x=0}^{x=N-1}\sum_{y=0}^{y=N-1}\delta(x,y)\cdot {W_N}^{x\cdot u+y\cdot v} && (5.7)\\
W_N &= e^{\frac{-2\cdot\pi\cdot j}{N}} && (5.8)\\
u &= \{0,1,...,N-1\} && (5.9)\\
v &= \{0,1,...,N-1\} && (5.10)
\end{aligned}$$

In the next sections we establish the relation between the detection threshold values $\tau_1$ and the minimum size and amplitude of deformation detectable with the probability of false detection ($P_f$) and the probability of not detecting a deformation ($P_n$). We also discuss the determination of the location threshold value $\tau_2$.

## 5.1 Probability of false alarm

In this case we are interested in the probability of the module of $R(u,v)$ to be bigger than our threshold value $\tau_1$, when there is not any deformation in the difference image, *i.e.* :

$$\delta(x,y) = \eta(x,y) \tag{5.11}$$

$$R(u,v) = Re(u,v) + j \cdot Im(u,v) \tag{5.12}$$

$$Re(u,v) = \frac{1}{N} \cdot \sum_{x=0}^{x=N-1} \sum_{y=0}^{y=N-1} \eta(x,y) \cdot cos(\frac{j \cdot \pi(x \cdot u + y \cdot v)}{N}) \tag{5.13}$$

$$Im(u,v) = \frac{1}{N} \cdot \sum_{x=0}^{x=N-1} \sum_{y=0}^{y=N-1} \eta(x,y) \cdot sin(\frac{-j \cdot \pi(x \cdot u + y \cdot v)}{N}) \tag{5.14}$$

Because each $\eta(x, y)$ is a gaussian distributed random variable, $Re(u, v)$ and $Im(u, v)$ are also gaussian distributed and, for N odd :

$$\mathbf{E}[Re(u,v)] = 0 \tag{5.15}$$

$$\mathbf{E}[Re(u,v)^2] = \begin{cases} 1 & \text{u=0 and v=0} \\ \frac{\sigma^2}{2} & \text{otherwise} \end{cases} \tag{5.16}$$

$$\mathbf{E}[Im(u,v)] = 0 \tag{5.17}$$

$$\mathbf{E}[Im(u,v)^2] = \begin{cases} 0 & \text{u=0 and v=0} \\ \frac{\sigma^2}{2} & \text{otherwise} \end{cases} \tag{5.18}$$

$$\mathbf{E}[Im(u,v) \cdot R(u,v)] = 0 \tag{5.19}$$

$$\tag{5.20}$$

The probability of false alarm $P_f$ can be found by integrating the joint distribution of $Re(u, v)$ and $Im(u, v)$ in a disk of radius $\tau_1$:

$$P_f = 1 - \frac{1}{\pi \cdot \sigma^2} \cdot \int_{-\tau_1}^{\tau_1} \int_{-\sqrt{{\tau_1}^2 - x^2}}^{\sqrt{{\tau_1}^2 - x^2}} e^{\frac{-(x^2+y^2)}{\sigma^2}} \quad dxdy \tag{5.21}$$

Therefore, for $u \neq 0$ and $v \neq 0$ :

$$P_f = e^{-(\frac{\tau_1}{\sigma})^2} \tag{5.22}$$

## 5.2 Probability of not detecting a deformation

In this case we want to find the probability of the module of $R(u, v)$ being smaller than our threshold when we have a deformation $f(x, y)$. As in the previous section, $Re(u, v)$ and $Im(u, v)$ are gaussian variables with the same variances given in the equations (5.16, 5.18, 5.19). However, if $F(u, v)$ is the DFT of $f(x, y)$, the means of $Re(u, v)$ and $Im(u, v)$ are now:

$$\mathbf{E}[Re(u,v)] = m_R \tag{5.23}$$

$$\mathbf{E}[Im(u,v)] = m_I \tag{5.24}$$

$$F(u,v) = m_R + j \cdot m_I \tag{5.25}$$

$$||F(u,v)|| = M = \sqrt{{m_R}^2 + {m_I}^2} \tag{5.26}$$

As before, the probability of not detecting a deformation can be found by integrating the joint distribution of $Re(u,v)$ and $Im(u,v)$ in a disc of radius $\tau_1$, which results in :

$$P_n = \frac{1}{\pi} \cdot \int_{\frac{1-\alpha}{\sigma} \cdot M}^{\frac{1+\alpha}{\sigma} \cdot M} x \cdot e^{-x^2} \cdot \arccos \frac{(1-\alpha^2) \cdot M^2 + x^2 \cdot \sigma^2}{2 \cdot x \cdot \sigma \cdot M} dx \quad (5.27)$$

$$\tau_1 = \alpha \cdot M \qquad 0 < \alpha < 1 \quad (5.28)$$

## 5.3 Window size and minimum amplitude of deformation

Equation (5.22) allow us to calculate the threshold value $\tau_1$ given the desired probability of false alarm and the noise level in the original range image. After $\tau_1$ is determined, using equation (5.27) we can determine the value of $||F(u,v)|| = M$ that will have probability $P_n$ of not being detected.

The value of $M$ is related to kind of deformation we want to detect, the size (area) occupied by the deformation and its maximum amplitude. In this paper we will consider periodic deformations corresponding to ridges, as shown in Figure 9. We chose this kind of deformation because it corresponds to a reasonable model of one kind of deformation produced during the milling process for manufacturing of sculptured surfaces. For this kind of deformation Figure 10 shows the relation between the value of $M$ and the area of the deformation in pixels (${N_d}^2$) when the maximum amplitude of the deformation is 0.1 mm. This graph was obtained by calculating the DFT of a synthetic deformation of amplitude 0.1 mm for different sizes of window $N_d$ and different values of deformation parameters (see Figure 9). The graph show the minimum value of $M$ obtained for each size of window.

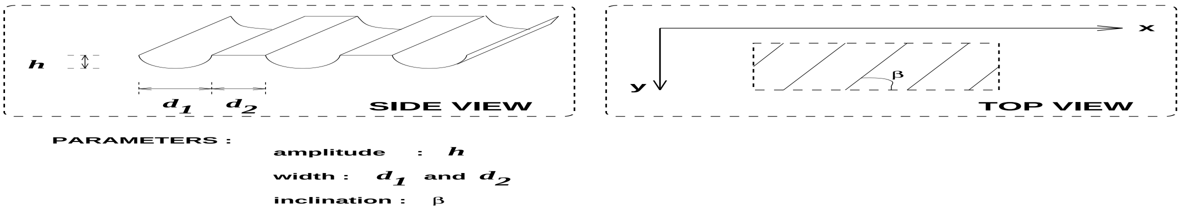


**Figure 9.** Ridges periodic deformation

It is important to remember that $N_d$ should be big enough to cover at least one spatial period of the deformation.

## 5.4 Estimating the position of the deformation

Assume that a deformation is detected inside a given observation window. The region of the deformation inside the window is localized by applying


**Figure 10.** Minimum module of DFT for periodic ridge deformation as function of $N_d$. Inclination varying from 0 to 90 degrees. Amplitude equal to 0.1 mm. Width varying from 3 to 10 pixels.

the inverse DFT to the selected peaks of the DFT of the residual in the observation window and then thresholding the results of the inverse DFT.

We now address the question of correctly locating true deformation points and not detecting random noise points.

The threshold value ($\tau_2$) used during the localization process is related to the probability of a point not belonging to the deformation being detected ($P_1$) and also to the probability of a point belonging to the deformation not being detected ($P_2$). In order to relate $\tau_2$ to $P_1$ and $P_2$ let us consider the reconstructed residual produced by the inverse DFT ($\hat{r}(x,y)$):

$$\hat{\delta}(x,y) = \frac{1}{N} \cdot \sum_{u=0}^{u=N-1} \sum_{v=0}^{v=N-1} K(u,v) \cdot (F(u,v) + N(u,v)) \cdot {W_N}^{-x \cdot u - y \cdot v} \quad (5.29)$$

$$\hat{\delta}(x,y) = \hat{f}(x,y) + \hat{\eta}(x,y) \quad (5.30)$$

$$\hat{f}(x,y) = \frac{1}{N} \cdot \sum_{u=0}^{u=N-1} \sum_{v=0}^{v=N-1} K(u,v) \cdot F(u,v) \cdot {W_N}^{-x \cdot u - y \cdot v} \quad (5.31)$$

$$\hat{\eta}(x,y) = \frac{1}{N} \cdot \sum_{u=0}^{u=N-1} \sum_{v=0}^{v=N-1} K(u,v) \cdot N(u,v) \cdot {W_N}^{-x \cdot u - y \cdot v} \quad (5.32)$$

$$K(u,v) = \begin{cases} 1 & \|F(u,v) + N(u,v)\| \geq \tau_1 \\ 0 & \text{otherwise} \end{cases} \quad (5.33)$$

$$(5.34)$$

As we can see in equation (5.29) the reconstructed residual is composed of two parts : a reconstruction of the deformation ($\hat{f}(x,y)$) and a reconstruction of the original noise in the residual ($\hat{\eta}(x,y)$). Because $\hat{\eta}(x,y)$ is a linear combination of gaussian variables, $\hat{\eta}(x,y)$ is also a gaussian variable and:

$$\begin{aligned}
\mathbf{E}[\hat{\eta}(x,y)] &= 0 && (5.35)\\
\mathbf{E}[\hat{\eta}(x,y)\cdot\hat{\eta}(x,y)] &= \frac{\sigma^2}{N^2}\cdot K_{ON} = {\sigma_{\hat{\eta}}}^2 && (5.36)\\
K_{ON} &= \sum_{u=0}^{u=N-1}\sum_{v=0}^{v=N-1} K(u,v) && (5.37)
\end{aligned}$$

From all these, we can conclude that the probabilities $P_2$ and $P_1$ for a given point $(x,y)$, such that $\hat{f}(x,y)=\rho$, is given by:

$$\begin{aligned}
P_1 &= 1 - \frac{1}{\sqrt{2\cdot\pi}\cdot\sigma_{\hat{\eta}}}\int_{-\tau_2}^{\tau_2} e^{\frac{-x^2}{{\sigma_{\hat{\eta}}}^2}}\,dx && (5.38)\\
P_2 &= \frac{1}{\sqrt{2\cdot\pi}\cdot\sigma_{\hat{\eta}}}\int_{-\tau_2-\rho}^{\tau_2-\rho} e^{\frac{-x^2}{{\sigma_{\hat{\eta}}}^2}}\,dx && (5.39)
\end{aligned}$$

Typically, the value of $K_{ON}$ will never be bigger than 30, therefore a reasonable upper limit to the standard deviation of $\hat{\eta}(x,y)$ $(\sigma_{\hat{\eta}})$ is:

$$\sigma_{\hat{\eta}} = 6\frac{\sigma}{N} \qquad (5.40)$$

## 6 An inspection example

As an example, if we consider a original range image with noise standard deviation $\sigma = 0.1mm$ and we want $P_f = P_n = 0.0001$ then from equations (5.22) and (5.27) we obtain : $\tau_1 = 0.31$ and $M = 0.55$. If we consider deformations as defined in Figure 9 with amplitude 0.1 mm the graph in Figure 10 says that the minimum deformation should cover a square of 50 pixels on a side in the range image. Because we do not know the exact position of the deformation, during the inspection procedure we used observation windows of 100 pixels, in such way that successive windows have half of their area in common.

From equation (5.40) the standard deviation of $\hat{\eta}(x,y)$ is equal $0.006mm$. Therefore if we use $\tau_2 = 0.05$, from equation (5.38) we obtain $P_1 \leq 2.2e{-}16$. Also, the probability of not detecting a deformation varies from $P_2 = P_1$ till $P_2 \sim 1$. In this case, assuming that the amplitude of the deformation is uniformly distributed. Thus, the expected value of $P_2$ is 0.5. These numbers mean that practically none of the points not belonging to the deformation will be detected and approximately 50% of the points belonging to the deformation will be detected.

Figure 11 shows the residual after registration between the test object number **1** model and a range image of the test object when a ridge deformation of amplitude 0.1 mm is artificially added to the data. Figure 12 shows the selected peaks of the DFT of the residual in a observation window over the deformed region. Finally, Figure (13) shows the deformed region detected by the inspection process in the residual using $\tau_1 = 0.31$ and $\tau_2 = 0.05$.

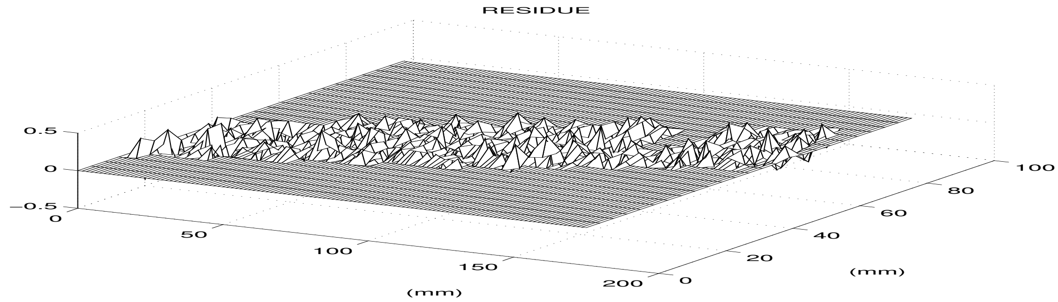


**Figure 11.** Residual after registration

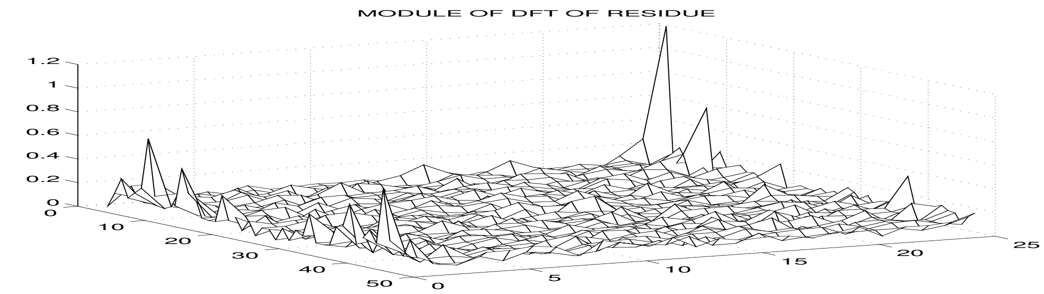


**Figure 12.** Module DFT of residual in an observation window over the deformation

## 7 CONCLUSIONS

We discussed the problem of inspecting free-form surfaces and presented some results concerning the use of B-spline models of free-form surfaces for the purposes of registration and inspection. Despite some initial promising results there are still many issues to be further investigated: evaluating the robustness to outliers of the modified version of the ICP, investigating further the monotonicity of the modified ICP algorithm, studying the use of

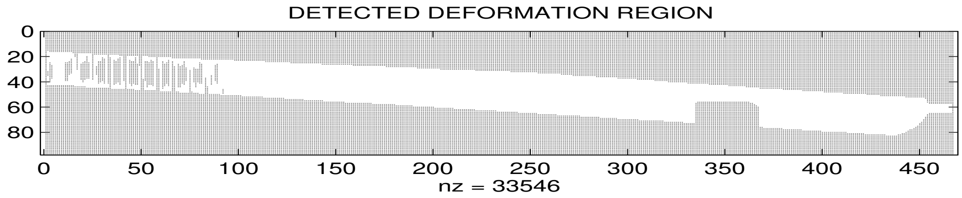


**Figure 13.** Final deformed region detected

techniques as the Hough Transform and search to obtain initial estimates of registration and, most of all, producing the final inspection diagnosis after the registration between model and data.

## 8 ACKNOWLEDGEMENTS

This work was supported by ACME under Grant GR/H/86905. We would like also to acknowledge the financial support of the CNPq in Brazil to Edvaldo Marques Bispo, and to thank Andrew Fitzgibbon, David Wren and the reviewers for their many suggestions and constructive criticizing.

## Bibliography

1. P. J. Besl: *Active optical range imaging sensors*, In J. L. C. Sanz, editor, *Advances in Machine Vision*. Springer-Verlag, New York, 1988.

2. D. K. Naidu, R. B. Fisher, G. Cameron: *User guide for the laser range image acquisition package*, University of Edinburgh, Dept. of AI, Imagine software paper no. 18, 1990.

3. C. DeBoor: *A practical guide to splines*, New York:Springer-Verlag, 1978.

4. K. S. Arun, T. S. Huang and S. D. Blostein: *Least-squares fitting of two 3-D point sets*, IEEE Trans. Patt. Anal. Mach. Intell., vol. PAMI-9, no. 3, Sept 1987.

5. B.K. Horn and J.G. Harris: *Closed-form solution of absolute orientation using unit quaternions*, J. Optical Society of America, vol. 4, no. 4, 1987, pp. 629 – 642.

6. O.D. Faugeras and M. Hebert: *The representation, recognition, and locating of 3-D objects*, Int. J. Robotics Research, vol. 5, no. 3, 1986, pp. 27 – 52.

7. C. Menq, H. Yau and G. Lai: *Automated precision measurement of surface profile in CAD-Directed Inspection*, IEEE Trans. Robotics and Automation, vol. 8, no. 2, April 1992.

8. D. Poussart and D. Laurendau, *3-D Sensing for Industrial Computer Vision*, In J. L. C. Sanz, editor, *Advances in Machine Vision.* Springer-Verlag, New York, 1988.

9. W. Grimson and T. Lozano-Perez: *Model-Based Recognition from Sparse Range and Tactile Data*, Int. Journal of Robotics Research, vol. 3, 1984, pp. 3 – 35.

10. P. J. Besl and N. McKay: *A Method for Registration of 3-D Shapes*, IEEE Trans. Patt. Anal. Mach. Intell., vol. PAMI-14, no. 2, Feb 1992, pp.239 – 256.

11. Y. Chen and G. Medioni: *Object modelling by registration of multiple range images*, Image and Vision Computing, vol. 10, no. 3 , April 1992.

12. N. P. Juster: *Modelling and representation of dimensions and tolerances: a survey*, Computer-Aided Design, vol. 1, no. 24, January 1992.

13. B. R. Suresh, R. A. F. Koushi, T. S. Levetti and J. E. Overlend: *A real-time automated inspection for hot steel slabs*, IEEE Trans. Patt. Anal. Mach. Intell., vol. PAMI-5, no. 6, Nov 1983.

14. M. Potmesil: *Generating models for solid objects by matching 3D surface segments*, Proc. Int. Joint Conf. on Artif. Intell., Karlsruhe, Germany, August 1983, pp 1089–1093.

15. A. A. G. Requicha: *Representation for rigid solids: Theory, methods and systems*, ACM Computing Surveys, vol. 12, no. 4, December 1980.

16. A. A. G. Requicha: *Toward a theory of geometric tolerancing*, The International Journal of Robotics Research, vol. 2, no. 4, winter 1983.

17. F. J. Stein: *Structural indexing for object recognition*, PhD thesis, IRIS, University of Southern California, April 1992.

18. Z. Zhang: *Iterative point matching for registration of free-form curves*, Research Report number 1658, INRIA, April 1992.