Int J Adv Manuf Technol (2000) 16:23–31
© 2000 Springer-Verlag London Limited

The International Journal of
Advanced
Manufacturing
Technology

CAD-Based Measurement Path Planning for Free-Form Shapes Using Contact Probes

I. Ainsworth, M. Ristic and D. Brujic

Mechanical Engineering Department, Imperial College of Science, Technology and Medicine, London, UK

Dimensional inspection of engineering components comprising free-form surfaces demands accurate measurement of a large number of discrete points, such that the actual shape may be fully characterised. This paper presents a methodology for CAD-based measurement of such components using a coordinate measuring machine equipped with a touch-trigger probe. The main shortcomings of the conventional methodology have been identified to be in relation to registration and probe radius compensation. The proposed measurement process involves the following main steps: registration, definition of measurement points, probe path generation, path optimisation and verification, measurement and probe radius compensation. By employing the CAD model at every step, the implemented methodology maximises the measurement accuracy and this is verified through a detailed simulation study. In addition, the implemented tools for CMM programming achieve accurate control of the overall measurement process and provide a high level of confidence when dealing with complex component geometry.

Keywords: CAD; CMM; Free-form shape; Measurement; NURBS; Registration

1. Introduction

This paper attempts to demonstrate an efficient, yet flexible, measurement planning methodology for components that consist partly or solely of free-form surfaces. The motivation for this work has been provided by the needs of modern manufacturing industry, particularly in the aerospace and the automotive sectors. Inspection of free-form components such as aero-engine turbine blades and car body panels involves the measurement of a sufficiently large number of points, with an appropriate point distribution, to enable a subsequent comparison to be made of the actual shape in relation to its nominal model. As part of this process, it is also becoming increasingly important to reconstruct the actual shape in a required CAD format [1] and to analyse the component performance using the available aerodynamic, structural and other computational analysis tools.

Computer controlled coordinate measuring machines (CMM), equipped with touch-trigger probes, represent the standard measuring instrument for dimensional inspection. In spite of a wide range of non-contact measuring systems becoming widely available, the CMM continues to be the first-choice solution for the metrology practitioners, owing to its high accuracy and the widely available operator skills [2]. Measurement accuracies of the order of 3–5 μm are readily achievable using modern CMMs.

Measurement of free-form shapes using a CMM and a contact probe poses considerable difficulties [3,4]. In order to capture the shape, it is necessary to perform dense measurements, where the required local point distribution is dictated by the local curvature, tolerances, and other factors. The commonly used "teach-by-showing" method for CMM programming is inadequate in such situations and it is necessary to employ CAD-based techniques for probe path generation, verification and collision avoidance [1,5]. Indeed, there are many commercially available packages for CAD-based CMM programming, such as CAMEO, SILMA, ICAMP, Origin and others. However, many of these packages are limited to performing "teach-by-showing" off-line, with the tools for automatic generation of measurement positions being limited in their ability to control adequately the local point density and to capture adequately the surface shape.

The accuracy achieved in performing measurements on free-form shapes using current methods and tools also causes concern. The root cause of these problems is the related issue of probe radius compensation and registration. Since the contact probe tip is a sphere of a given radius, the raw measurement data, represented by the ball centre positions, must be compensated by introducing a correct offset in the direction of the surface normal at the point of contact. Current methods for probe radius compensation are based on the assumption that the probe makes contact with the surface at exactly the prescribed point, so that the surface normal vector is known. However this assumption is generally not true in the presence

Correspondence and offprint requests to: I. Ainsworth, Imperial College of Science, Technology and Medicine, Mechanical Engineering Department, Exhibition Road, London SW7 2BX, UK. E-mail: i.ainsworth@ic.ac.uk

of any errors in registration, leading to incorrect compensation and inaccurate final results.

Registration, which establishes the relation between the CMM coordinate frame and that of the CAD model, poses considerable difficulties because free-form objects generally do not possess clearly identifiable reference features [6–8]. The available measurement packages prescribe the registration procedure to be performed on the basis of adequately compensated measurement data. However, since accurate compensation requires accurate registration as a prerequisite, considerable errors may be introduced at this stage.

In addition, there are further sources of registration error that are often introduced in practice. Many commercial CMM packages assume that for each measured point, its corresponding point on the CAD model is known *a priori*, but this is never strictly true. Furthermore, the prevailing approach is to employ only 20–30 points, at most, for registration, because of the difficulties in collecting large data sets at the registration stage and also because of the computing time required by some least-squares fitting algorithms. However, it has been conclusively shown [6] that such a small number of points can lead to considerable registration errors in many situations.

The CAD-based measurement planning methodology proposed in this paper was designed to address these issues through a combination of appropriate methods for measurement planning, registration and probe radius compensation. As the nominal CAD model is assumed to be represented by NURBS (non-uniform rational B-spline), Section 2 provides a brief overview of NURBS definition and of some necessary NURBS computations. Section 3 deals with registration and probe radius compensation, showing how offset nominal NURBS entities can be employed to achieve accurate results. Section 4 deals with the definition of the measurement points, where the number of measurements and their distribution are controlled by a number of relevant factors. The methods implemented for automatic probe path generation are presented and discussed in Section 5, including the implementation of measurement simulation, as the means of verifying the probe path prior to execution. Finally, in Section 6 we propose a practical method for automatic collision detection which was designed to complement graphical visualisation tools and we describe its implementation.

2. Definition and Manipulation of NURBS Surfaces

NURBS [9] were chosen as the main modelling geometric entity because they are supported by most modern CAD/CAM systems and data exchange standards (IGES, STEP). Furthermore, NURBS also provide exact modelling of natural quadric shapes, such as cones and cylinders, making the proposed methodology also applicable to such shapes.

2.1 NURBS Definition

A NURBS surface of degree p in the u-direction and degree q in the v-direction is a bivariate vector-valued piecewise rational function of the form

$$S(u,v) = (x_s(u,v), y_s(u,v), z_s(u,v)) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)w_{i,j}} \qquad (0 \le u,v \le 1) \tag{1}$$

where the control points $\{\mathbf{P}_{i,j}\}$ form a bi-directional control net, and $\{w_{i,j}\}$ are control point weights. The functions $\{N_{i,p}(u)\}$ and $\{N_{j,q}(v)\}$ are the non-rational B-spline

$$U = \{\underbrace{0,\ldots,0}_{p+1}, u_{p+1},\ldots,u_{r-p-1}, \underbrace{1,\ldots,1}_{p+1}\}$$

$$V = \{\underbrace{0,\ldots,0}_{q+1}, v_{q+1},\ldots,v_{s-q-1}, \underbrace{1,\ldots,1}_{q+1}\} \tag{2}$$

basis functions defined on the knot vectors where $r = n + p + 1$ and $s = m + q + 1$.

By introducing the piecewise rational basis functions

$$R_{i,j}(u,v) = \frac{N_{i,p}(u)N_{j,q}(v)w_{i,j}}{\sum_{k=0}^{n}\sum_{l=0}^{m} N_{k,p}(u)N_{l,q}(v)w_{k,l}} \tag{3}$$

the surface equation (1) can be written as

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} R_{i,j}\,(u,v)P_{i,j} \tag{4}$$

Equations (1–4) define evaluation of a point on a NURBS surface and this function was implemented as outlined in [10].

2.2 NURBS Surface Tangents and Normals

In order to find the normal direction at an arbitrary point on a NURBS surface, the tangential direction in u and v must first be computed using the following equations.

$$\mathbf{T}_u(u,v) = \frac{\partial}{\partial u}\,\mathbf{S}(u,v) \quad \text{and } \mathbf{T}_v(u,v) = \frac{\partial}{\partial v}\,\mathbf{S}(u,v) \tag{5}$$

The surface normal is then calculated as the cross-product of the two tangent vectors:

$$\mathbf{N}(u,v) = \mathbf{T}_u(u,v) \times \mathbf{T}_v(u,v) \tag{6}$$

Equations (5) and (6) were implemented according to Peterson [11]. However, additional modifications were made in collaboration with Peterson, in order to improve the calculation of the magnitude of the partial derivatives.

2.3 Line and NURBS Surface Intersection

In order to determine the point of intersection between a line (starting at a point P_0, direction vector V) and a NURBS surface, we have to solve the system of equations comprising the NURBS surface (Eq. (1)) and the line equation:

$$P(t) = (x_P(t), y_P(t), z_P(t)) = P_0 + t \times V \tag{7}$$

As the solution to this system of equations cannot be found in a closed form, a standard iterative procedure was used.

First, the functional relations defining the distances between two arbitrary points, one on the surface $S(x, y, z)$ and one on the line $P(x, y, z)$, are introduced as follows:

$$f_x = x_S(u,v) - x_P(t) \tag{8}$$

$$f_y = y_S(u,v) - y_P(t) \tag{9}$$

$$f_z = z_S(u,v) - z_P(t) \tag{10}$$

Bearing in mind that x_S, y_S, and z_S are functions of parameters u and v, whereas x_P, y_P and z_P are functions of the parameter t, each of the functions f_i can be expanded into a Taylor series in the neighbourhood of u, v, and t in the standard fashion:

$$f_i(u + \delta u, v + \delta v, t + \delta t) = f_i(u,v,t) + \frac{\partial f_i}{\partial u} \times \delta u + \frac{\partial f_i}{\partial v} + \delta v + \frac{\partial f_i}{\partial t} \times \delta t + O(\delta u^2) + O(\delta v^2) + O(\delta t^2) \tag{11}$$

By neglecting terms of the second and higher orders, and by setting

$$f_i(u + \delta u, v + \delta v, t + \delta t) = 0 \tag{12}$$

we obtain a set of linear equations for the corrections δu, δv, δt that simultaneously move each function towards zero:

$$\mathbf{J} \times \delta\mathbf{X} = -\mathbf{F} \tag{13}$$

where $\mathbf{J}$ is the Jacobian matrix, $\delta\mathbf{X} = \{\delta u, \delta v, \delta t\}^T$ and $\mathbf{F} = \{f_x, f_y, f_z\}^T$. Starting with an initial guess we apply a number of Newton–Raphson steps to improve the solution. We stop if the solution converges in either summed absolute variable increment or summed absolute function values.

The initial guess is critical to the method. A satisfactory initial guess for the point on the line may be obtained from the geometry of the problem, and it is in most cases provided by the starting point P_0. The initial guess for the surface point, however, is not found in a straightforward manner. First, the surface was approximated by a dense triangular mesh. Then, all the triangles were tested for intersection with the line [12]. All of the line–triangle intersections are then interrogated and the one found to be the closest to the initial point on the line is recorded. The vertex closest to the recorded intersection point was found to be a satisfactory initial guess for the point on the surface.

2.4 Offset NURBS Surface

Much of the work that follows makes use of offset NURBS surfaces. An offset surface $\mathbf{O}(u, v)$ is specified by:

$$\mathbf{O}(u,v) = \mathbf{S}(u,v) + d\mathbf{N}(u,v) \tag{14}$$

where d is the offset distance. It has been proved [9] that, given a NURBS surface $\mathbf{S}(u, v)$, its offset $\mathbf{O}(u, v)$ is generally not a NURBS. Therefore evaluation of an offset NURBS surface implies a degree of approximation.

In our implementation, the offset surface is calculated using linear least-squares fitting [13]. The method first samples the original surface in the u and v-directions, producing a regular grid. The minimum number of samples that has to be taken in order to construct the offset surface is one point per knot span, but it was found that using three points per knot span gave a good balance between speed and accuracy. Each sample point is then projected by a distance d in the direction normal to the surface. Once all of the offset points have been generated, the offset surface can then be fitted in a least-squares fashion. The parameterisation for the new surface is taken directly from the original one. However, in some situations it is necessary to perform knot insertion in order to increase the flexibility of the surface in areas of high curvature. Regions where knot insertion is required may be identified on the basis of a user-specified tolerance value.

3. Registration

Since the measurements on the object are made relative to the coordinate frame of the CMM, but the nominal CAD model is defined relative to some arbitrary coordinate frame, it is necessary to establish the relationship between the two frames through registration [6,7]. In principle, registration evaluates the transformation (rotation and translation) that must be applied to the nominal model in order to align it with the actual object. Accurate registration enables the CMM to negotiate the correct path around the object and also provides the basis for subsequent error analysis.

Conventional registration techniques generally follow one of two approaches. First, if the component has three orthogonal planes as a reference feature, then such a component can be readily registered using six points (3, 2, and 1 on each plane, respectively). Secondly, if such features are not available, the method is to identify and subsequently measure six or more arbitrary points, each corresponding to a point defined on the model [7]. Using the information about the corresponding points allows the transformation matrix to be found.

However, many free-form components do not possess suitable reference features and the application of the 3–2-1 registration procedure implies development of special purpose reference jigs, which would hold the component and provide the necessary datum. Although common in practice, this solution is expensive and inflexible. Furthermore, in the application of the second method, it is difficult to guide the CMM to measure at precisely the defined corresponding points. Thus, a detailed procedure for establishing point correspondences must be developed specifically for each new shape, again resulting in an inflexible and costly solution.

3.1 Proposed Registration Method

The principle of the proposed solution for the registration problem involves the least-squares alignment of the raw measurement data (sphere centre positions) with surface entities that are offset by the length of the ball radius from the nominal model. This is in direct contrast to the conventional methodology of using the compensated data and the nominal surfaces.

As Fig. 1 illustrates, the rationale behind this solution lies in the fact that the probe sphere centres, represented by the raw measurement data, inevitably lie on a surface which is

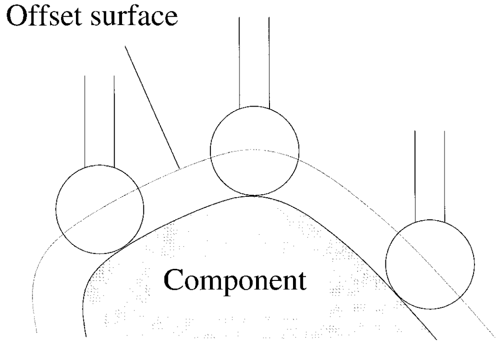

Fig. 1. Offset surface registration.

offset from the object by the sphere radius. Hence, this offset surface may be readily employed for registration purposes, even if the actual points of contact between the sphere and the object are not known.

Following this, registration was realised through the implementation of the ICP (iterative closest point) method, which was reported in [8] and verified in [6]. As the name suggests, the ICP method minimises, at each iteration step, the collective square distances between the measured points and their closest points on the surface. The ICP algorithm was shown to converge in the presence of measurement noise and for reasonable initial misalignments of several millimetres in position and several degrees in orientation. Importantly, the implemented algorithm was shown to be highly efficient when using NURBS, performing registration of data sets containing many thousand points in a matter of minutes.

However, when the model and the part are initially grossly misaligned, convergence of the ICP algorithm cannot be guaranteed. For this reason, it is necessary to bring the two closer together by some other means, as the first alignment step. In our work this was readily achieved through a slight variant of the conventional corresponding point method. Simply, this means that six or more points are identified on the nominal model and then measured on the actual object by driving the CMM manually. The corresponding points on the offset nominal surface are then aligned to the measured raw data in a least-squares fashion. Clearly, the objective in this step is to align the two shapes only approximately, so it is not important that the two point sets correspond exactly. In our experience with a variety of engineering parts, this approach was found to produce sufficiently good initial alignment quickly, allowing full subsequent ICP registration to proceed.

It is also worth noting that the registration accuracy will be greatly increased if a large number of points is used for registration [6]. We therefore suggest that a larger number of points be used for ICP registration than for the initial alignment and that this process be repeated once the full measurement data set is obtained, in order to maximise the overall registration accuracy.

3.2 Validation of the Proposed Registration Method

Intuitively, the proposed method may be expected to produce superior results compared with the conventional techniques because it does not rely on strict point correspondences and avoids the introduction of any errors due to probe compensation. In order to verify the method, however, we conducted a Monte Carlo simulation and analysed its performance in relation to a conventional approach based on compensated measurements.

Figure 2 shows the model of the component that was used to validate registration, which is effectively a vertex of a cube. This shape was chosen because it provides the most direct relationship with the conventional registration methodology, which uses three orthogonal flat surfaces as the registration feature. Measurements were simulated by sampling a set of 900 points and introducing an offset of 1 mm in the direction of the surface normal to represent the probe offset. No measurement noise was simulated. The model was then transformed randomly 30 times. The transformation was composed of translation and rotation, both values of which were generated as uniformly distributed random numbers within the range of ±1 mm and ±1°, respectively. Misalignment of this magnitude was considered to be representative of practical situations, when an approximate object position can be established relatively easily. At the same time, such a small misalignment ensured convergence of both registration methods, overcoming the known degeneracies when performing least-squares fitting on a cube.

For each simulation set, registration was applied using the proposed offset surface method and the method which uses compensated data, where the compensation was calculated along the surface normals of the unperturbed model. The registration errors were computed as the differences between the initial perturbation and the results of the registration procedures. Table 1 presents statistical analysis of the registration errors for the two methods, comparing the standard deviation and mean values of registration error. The results show that the offset surface method produces consistently higher accuracy of alignment owing to the absence of compensation errors. Table 1 also presents the 95% confidence intervals for these experiments, further confirming the superior performance of the proposed registration method.

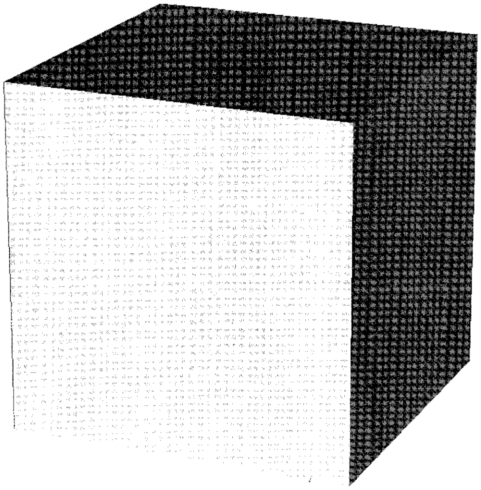

Fig. 2. Model used for registration.

Table 1. Results of the Monte Carlo simulation for the registration process.

	Standard deviation of the registration error					
	Rot *X* (deg)	Rot *Y* (deg)	Rot *Z* (deg)	Trans *X* (mm)	Trans *Y* (mm)	Trans *Z* (mm)
Offset surface method	0.0014	0.0004	0.0020	0.0084	0.0095	0.0121
Conventional compensation	0.1783	0.0820	0.0676	0.0167	0.0239	0.0166
	Mean value of the registration error					
	Rot *X* (deg)	Rot *Y* (deg)	Rot *Z* (deg)	Trans *X* (mm)	Trans *Y* (mm)	Trans *Z* (mm)
Offset surface method	–0.0001	0.0001	–0.0006	0.0037	–0.0006	0.0032
Conventional compensation	0.0153	0.0268	–0.0152	–0.0100	–0.0220	–0.0193
	95% registration accuracy confidence level					
	Rot *X* (deg)	Rot *Y* (deg)	Rot *Z* (deg)	Trans *X* (mm)	Trans *Y* (mm)	Trans *Z* (mm)
Offset surface method	0.0009	0.0003	0.0012	0.0052	0.0059	0.0075
Conventional compensation	0.1105	0.0508	0.0419	0.0103	0.0148	0.0103

4. Definition of Measurement Points

The required distribution of the measurements is determined by the shape of the object in question, the type of defect that is of interest and by the requirements for subsequent processing and analysis of the data. The required local measurement density is primarily dictated by the local surface curvature. This may be determined on the basis of the nominal CAD model and it may be further increased to account for the type of surface defect that is of interest. Bearing in mind that real data always contain noise, the surface sampling density may then be further increased to facilitate subsequent filtering of the measurement noise.

A conventional way to define discrete measurement points on a measured entity is to define a 2D rectangular grid (parallel to one of the primary planes) within the 3D model environment. Once the grid has been positioned and the number of points on the grid specified, the grid points are then projected on the model surface (see Fig. 3). Each point of intersection between the projection lines and the model will specify a measurement point to be taken during the measurement cycle. It can be seen that in some simple situations this method of specifying the measurement points would work reasonably well. However, trying to use this method to measure more complex components is very difficult. It is particularly difficult, and often impossible, to position the grid such that it will produce good coverage over a surface with high curvature or large aspect ratio. Therefore, if surfaces such as these were to be measured, then it would have to be carried out in stages. This would further complicate the procedure and incur considerable time and cost penalties.

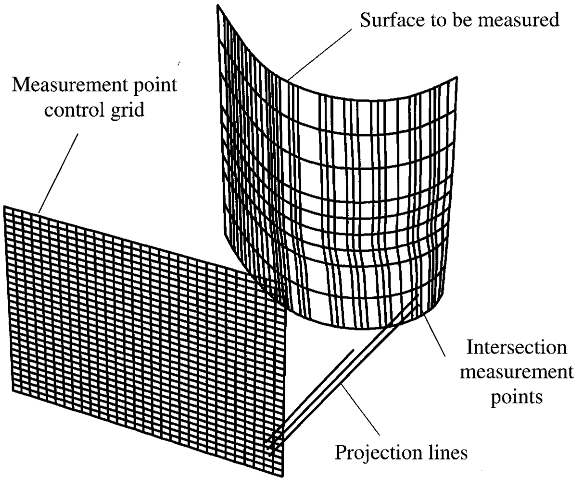
Surface to be measured
Measurement point control grid
Intersection measurement points
Projection lines

Fig. 3. Conventional measurement point generation.

4.1 Proposed Methodology

A typical nominal model of a component would constitute one or more NURBS surfaces. To present the most general case, the discussion will focus on measuring one single NURBS surface. However, the methodology implemented will cater for multiple surfaces or smaller regions within a single surface.

As already mentioned, the sampling of the surface of interest must be such that it produces the required measurement point distribution. The adopted method involves adaptive subdivision sampling of the nominal model entities, in order to ensure the required coverage according to local surface topology and regardless of its orientation.

Several surface sampling criteria have been identified and implemented, to be employed by the user individually or in a combination. These are as follows:

Uniform sampling in the u and v parametric directions. This is the simplest criterion, resulting in the surface being broken down into a rectangular grid, with even *u*-*v* spacing across the whole of the surface [14].

Chord length criterion. Local sampling density is controlled by specifying the maximum chordal deviation, this being the largest distance between a line connecting any two adjacent points and the surface. The resulting point density is therefore dependent on the local surface curvature [14] and, as Fig. 4 shows, this criterion allows a higher point density at locations of high curvature.

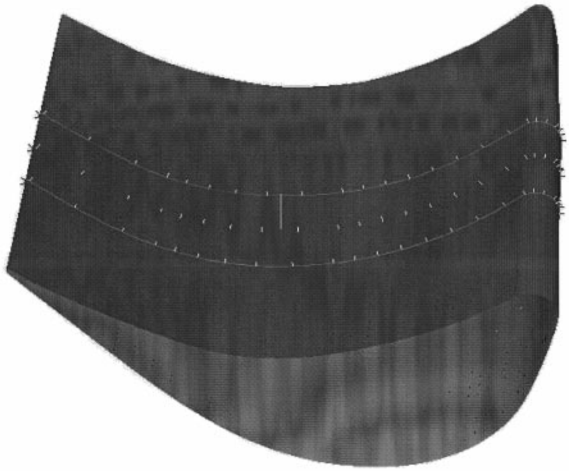

Fig. 4. Point generation based on local curvature.

Minimum sample density criterion. This criterion specifies the maximum allowed distance between any two neighbouring points. This was found to be necessary because the chord length criterion will often result in relatively flat surface regions being sampled at an undesirably low density. An example of the chord length criterion combined with minimum sample density can be seen in Fig. 5.

Parameterisation-based sampling criterion. Least-squares fitting of a NURBS surface through point data (such as that employed to find offset surfaces, Section 2) requires that there is at least one point situated within each knot span. This sampling criterion is therefore based on the model parameterisation, with the required number of samples per knot span being specified by the user.

Following this, the recursive subdivision sampling algorithm may be summarised as follows:

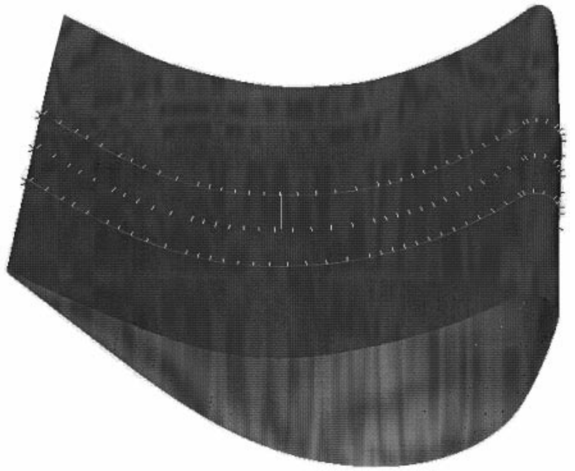

Fig. 5. Point generation based on local curvature, with maximum distance in *u* set to 2 mm.

```
for u and v parametric directions
{
  – sample the start point and end point of the surface.
    – examine the segment for prescribed criteria (e.g.
      chordal error)
      if criteria is not within tolerance {
        SubDivide(start point, end point)
      }
    }
}
SubDivide(previous point, next point)
{
  – sample another new point half way between the
    previous point and the next point.
  – examine the segment between previous point and
    new point for prescribed criteria.
  if criteria is not within tolerance {
    SubDivide(previous point, new point)
  }
  – examine the segment between new point and next
    point for prescribed criteria.
  if criteria is not within tolerance {
    SubDivide(new point, next point)
  }
}
```

The algorithm therefore ensures that the local sampling density is never lower than that prescribed by the above criteria.

5. Probe Path Generation, Modification and Verification

Generating the probe path involves producing all the machine instructions necessary to guide the CMM hardware through the measurement cycle. It should be performed in such a way that enables the machine to negotiate each measurement point, whilst ensuring that there are no areas of collision between the probe sensor and the component. The CMM used in this work was an LK G-90C, equipped with a Renishaw PH-10 indexing head and a Renishaw TP2–5W touch trigger probe.

5.1 Probe Path Generation

There are a number of issues to consider in order to achieve an efficient probe path. Users must be provided with programming tools that enable them to exercise their judgement and control all aspects of the measuring process. Much work has been undertaken in order to automate this process [3,15]. However, we take the approach that there must be interactivity between the CAD system and the user, if an optimal probe path is to be generated. In our work, we adopted the approach that the model entities define the regions of the object to be measured, and that the measurement should be performed for each entity in turn. In some situations it may be desirable that two or more entities be measured as a single surface and this can be readily achieved by stitching adjacent NURBS surfaces into one. This functionality is readily available in modern CAD/CAM systems.

The order in which the measurement points are negotiated must be adapted to the geometry in question. With each entity being essentially sampled over a grid of points, the

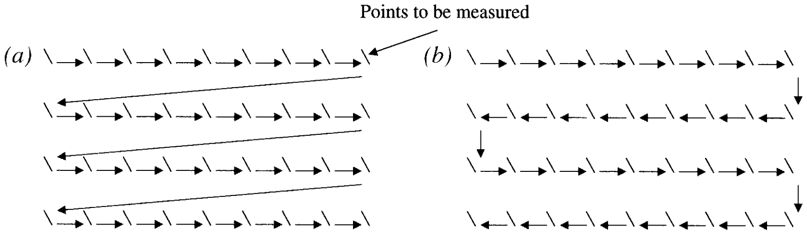

Fig. 6. (*a*) Unidirectional path. (*b*) Bidirectional path.

measurements may be performed in unidirectional or bi-directional scans (Fig. 6), where the former is generally better suited to closed and/or highly folded surfaces (such as a turbine blade airfoil), and the latter is more suited to relatively flat, open surfaces. It is also desirable to optimise the path in order to reduce the number of probe head re-orientations, since re-orientation is a relatively time-consuming operation, and the CMM must be calibrated for each orientation used. Finally, as each probing cycle of a touch-trigger probe involves a very low approach speed, there is a need to reduce the stand-off distance in order to optimise the overall measuring time. However, reducing the stand-off distance increases the risk of collision between the probe and the part, and this demands careful verification of the generated path.

Using the CAD model and the generated sampling points as the input, the implemented path planning software initially generates a measurement path for each selected entity, based on the default parameters set by the user. These include the scan direction (*u* or *v*), scanning pattern (unidirectional or bi-directional) stand-off distance, probe head orientation and others. The path is displayed as a set of line segments, together with the 3D model of the part. Following this, the system allows the user to modify interactively any of the path parameters. The user may also select any of the segments and modify the stand-off distance and probe orientation. Segments may be deleted or edited to introduce additional "via points" through which the probe will pass when executing point-to-point moves and/or re-orientation.

Finally, the defined measurement path is post-processed into machine executable programming code. We adopted DMIS (dimensional measurement interface standard) as the machine instruction language, being the most widely supported standard for CMM programming.

5.2 Probe Path Verification

Verification of the machine movements, prior to downloading instructions to the CMM, is critical for providing the user with full confidence in the validity of the generated measurement path. This was realised through implementation of a simulation and a full 3D graphical animation [16–18] of the measuring process. The input for the simulation is provided by the DMIS code produced by the path generation module. The code is interpreted and executed to emulate the operation of the actual CMM. This approach also allows simulation of externally generated CMM programs.

It was decided that the animation should provide full 3D rendering, as opposed to wire-frame graphics, in order to ensure that the depth of the scene and the probe/part interaction are always comprehensible. The animation is dynamic, allowing the user to zoom and rotate the scene while the animation is running. Any part of the animation can be rewound and replayed, if it is deemed to be of particular importance. The complete animation can be controlled using a slider bar, allowing the operator to rapidly move through the measurement cycle. Hence, using this option the operator could easily verify a path containing thousands of movements. Having this kind of facility was found to be of immense practical value, allowing the operator to validate the measuring cycle visually and to identify any unforeseen problems, such as collisions between the probe and the part.

Producing fully rendered animations of scenes containing NURBS entities is a computationally intensive task. The graphics library used to develop the software was OpenGL. Figures 7 and 8 are still frames of an animation that runs in real-time on standard Pentium II PC hardware platforms that do not have any specialised graphics acceleration. The required computational efficiency in generating the animations was achieved by generating a triangulated model of the part before passing it to OpenGL for rendering, rather than passing the NURBS entities. In generating the triangulated model, the original NURBS entities are sampled according to the criteria based on the local surface curvature, as outlined in Section 4, to produce a model that is accurate to within a specified tolerance.

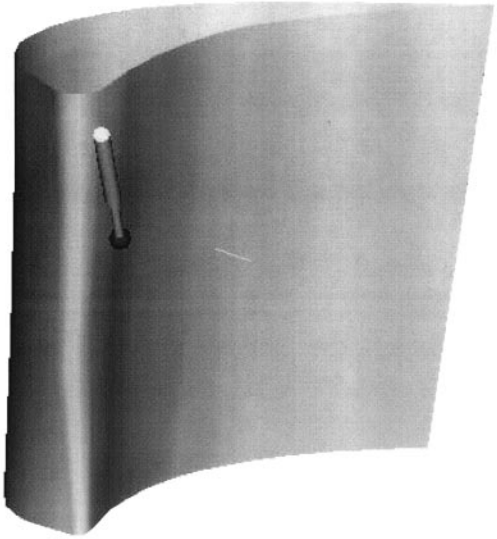

Fig. 7. Probe path verification.

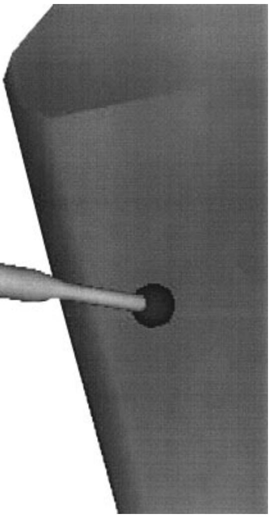

Fig. 8. Probe path verification, alternative view.

The quality of the animated scene was further improved by the implementation of smooth shading of the polygon segments.

6. Automatic Collision Detection

Visual examination of the simulated probe path is adequate in many cases, in particular when examining for gross path errors, such as collision of the probe head with the object and its surroundings. However, in situations involving dense measurements, in the vicinity of small features and with a small standoff distance, it was found that visual path verification is still difficult. In such cases, a collision detection tool that detects collisions with the object and highlights the offending path segments is highly beneficial [19,20].

The collision detection algorithm considers collision between the probe tip and the nominal model, instead of examining each probe movement for collision between the probe sphere and the surface of the model. Considerable gains in computational efficiency were realised by examining for intersection between the trajectory of the sphere centre and the offset model surfaces (the offset being the sphere radius). Figure 9 shows a schematic of this solution. The point of intersection was found using the method outlined in Section 2.3. As discussed, the starting point of each segment provided the initial guess for the point on the line, while the initial guess for the point on the surface was derived from its triangulation. These initial guesses have always provided convergence of the Newton–Raphson method.

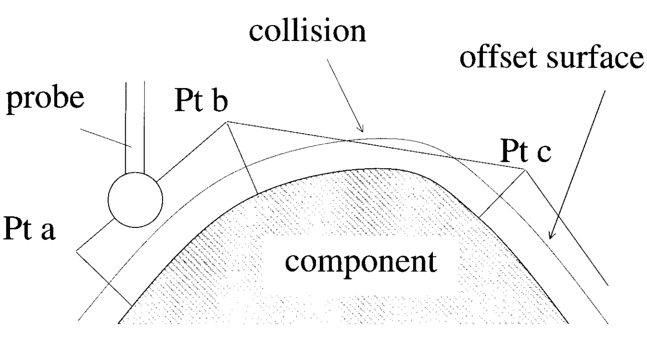

Fig. 9. Automatic collision detection.

7. Conclusion

The paper has presented an implementation of a methodology for measurement planning and collision-free probe path generation to perform measurements on a free-form shape using a CMM and a touch-trigger probe. The method uses the CAD model of the part at each step, with NURBS being the principal modelling entity. Offset surface entities were employed in order to minimise the errors that may result from probe radius compensation and to improve the computational efficiency in carrying out certain tasks. Accurate component registration was identified as a key requirement and the performance of the proposed registration method was demonstrated. Adaptive subdivision sampling according to several relevant criteria was employed to define the required measured points. Interactive graphic tools for probe path generation, modification and verification were developed. Fully rendered computer animation of the measurement process and the automatic collision detection are considered to be necessary practical tools and a key to providing the user with confidence in the generated results. The methodology and the system have been applied to measurement of various engineering parts, including aeroengine turbine blades, whose dimensional inspection is generally difficult to perform.

References

1. C. Menq and F. L. Chen, “Curve and surface approximation from CMM measurement data”, Computers and Industrial Engineering, 30(2), pp. 211–225, 1996.
2. Hong-Tzong Yau and Chia-Hsiang Menq, Path Planning for Automated Dimensional Inspection Using Coordinate Measuring Machines, Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, California, April 1991.
3. H. T. Yau and C. H. Menq, Automated CMM path planning for dimensional inspection of dies and moulds having complex surfaces. International Journal of Machine Tools and Manufacture, 35(6), pp. 861–876, June 1995.
4. C. P. Lim and C. H. Menq, CMM feature accessibility and path generation. International Journal of Production Research, 32(3), pp. 597–618, 1994.
5. B. Shirinzadeh, “Task planning and offline programming”, Industrial Robot, 23(5), pp. 4–5, 1996.
6. D. Brujic and M. Ristic, “Analysis of free form surface registration”, Proceedings Institution of Mechanical Engineers, 211, Part B, pp. 605–617, 1997.
7. K. C. Sahoo and Chia-Hsiang Menq, “Localization of 3-D objects having complex sculptured surfaces using tactile sensing and surface description, Journal of Engineering for Industry, 113, pp. 85–92, February 1991.
8. M. Ristic and D. Brujic, “Efficient registration of NURBS geometry”, International Journal of Image and Vision Computing, 15, pp. 925–935, 1997.
9. Les Piegl and Wayne Tilller, The NURBS Book, 2nd edn, Springer, 1997.
10. R. H. Bartels, J. Beatty and B. Barsdy, “An introduction to splines for use in computer graphics and geometric modeling”, 21(4), pp. 103–110, 1987.

11. John W. Peterson, Tessellation of NURBS Surfaces, Graphic Gems IV, Paul S. Heckbert, Academic Press, pp. 287–319, 1994.
12. D. Badouel, An Efficient Ray-Polygon Intersection, Graphic Gems II, James Arvo, Academic Press, 1990.
13. Weiyin Ma and J. P. Kruth, "Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces", Computer-Aided Design, 27(9), pp. 663–675, 1995.
14. H. J. Pahk, M. Y. Jung, S. W. Hwang, Y. H. Kim, Y. S. Hong and S. G. Kim, "Integrated precision inspection system for manufacturing of moulds having CAD defined features", International Journal of Advanced Manufacturing Technology, 10(3), pp. 198–207, 1995.
15. E. Lu, J. Ni and S. M. Wu, "Algorithm for the generation of an optimum CMM inspection path", Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, 116(3), pp. 396–404, September 1994.
16. S. Kochhar and J. Hall, "Unified, object-oriented graphics system and software architecture for visualising CAD/CAM presentations", Computer Graphics Forum, 15(4), pp. 229–248, October 1996.
17. R. J. Hollands and N. Mort, "Manufacturing systems simulation with enhanced visualisation using virtual reality techniques", IEE Colloquium (Digest), 166-MD, pp. 6/1–6/4, 1995.
18. S. Zeghloul, B. Blanchard and M. Aurault, "SMAR: a robot modeling and simulation system", Robotica, 15(1), pp. 63–73, January–February 1997.
19. C. H. Fedrowitz, "Fast intersection detection algorithm for PC-based robot off-line programming", Proceedings of SPIE – The International Society for Optical Engineering, 2247, pp. 152–162, 1994.
20. C. J. Wu, "On the representation and collision detection of robots", Journal of Intelligent and Robotic Systems: Theory and Applications, 16(2), pp. 151–168, June 1996.