

# **A Methodology to Create STL Files from Data Point Clouds Generated with a Coordinate Measuring Machine**

Irenee Tyvaert, Universite de Technologie de Troyes, France  
Prof. Georges Fadel, Clemson University  
Prof. Emannelle Rouhaud, Universite de Technologie de Troyes, France

## **Abstract**

Industries have always had the need to reproduce objects especially when drawings are no longer or were never available (reverse engineering). Coordinate measurement machines, whether touch- or laser-based, enable users to generate a point cloud representing a surface. The aim of this work is to develop a methodology and its translation into a software program to generate closed STL file models from such data point clouds.

The first step in the research is to develop an algorithm to merge several surfaces (data point clouds) from a real object in exactly the same reference frame. Then, a tessellation program is implemented to create the STL file considering computational costs and minimizing the number of triangles while maintaining the accuracy of the original part.

The implementation is illustrated on two examples and conclusions are drawn.

Keywords: Reverse Engineering, Rapid Prototyping, Tessellation, CMM, and STL

## **Introduction**

Reverse engineering is not a novel idea. It was probably born out of the need to reproduce either virtually or physically some human organs for medical applications. In fact it is difficult to accurately draw bones or organs with CAD software; accurate models from in-situ organs were represented in 2D slices obtained from CAT Scan images [1]. Physicians wanted to be able to test surgical procedures before operating on a patient, and thus needed an accurate three-dimensional representation of the local environment. Reverse engineering was what they hoped for. Mechanical industries also had a vested interest in such a technology because some CAD models were often unavailable or unusable for parts, which had to be duplicated or modified [3]. With the advent of coordinate measuring machines (CMM), the ability to measure a part and obtain a point cloud became reality. That point cloud was and still is mainly used for measurement and validation purposes. However these machines use contact probing digitization of a part's surface and thus cannot work on some fragile parts. Furthermore, they are relatively slow, and require careful programming to accomplish the desired task. Non-contact sensing techniques emerged later, mostly laser based. Those laser techniques rely on a laser beam projected onto the part's surface while optical sensors receive the reflected beam. The digitization is very fast and does not depend on surface contact. Unfortunately, the results are affected by the reflective quality of the surface and shadows or scratches on the surface can have a significant effect on measurement accuracy. Moreover, the price of these machines is important. For these reasons, and because many industries have a CMM in house to control the quality of their production, the CMM is the main source of data considered in this paper for reverse engineering applications.

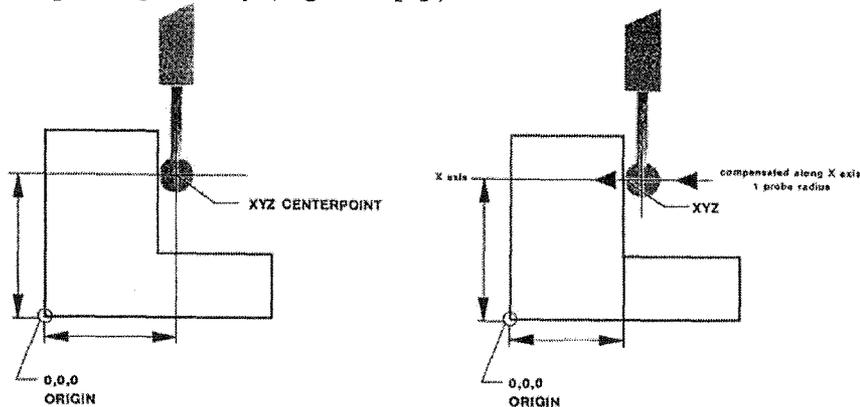
Rapid prototyping is a technology that has evolved to enable users to produce directly a physical artifact from CAD data. For this research, our goal is to enable a link between a coordinate measuring machine and a rapid prototyping machine, thus realizing in effect a three-dimensional duplication machine (stereo-copy). This technology uses the STL format; its generation is easier than any other standard CAD formats (IGES, STEP). An STL file contains only coordinates of vertices in 3D space that are sorted to form triangles and to express on which side of the triangles the material lies (surface normal).

### Coordinates Acquisition with a CMM

Several parameters control the quality of the coordinates acquired in a CMM. In this section, we describe some of the machine characteristics. The CMM machine used in our implementation is a Brown and Sharpe MicroVal PFX with the Micromasure IV version 3.1 control software.

#### *Probe type:*

The probe type and geometry are key to the quality of the measurements. The selection of the most appropriate probe is usually based on experience. If the probe tip is too big, namely larger than the radius of curvature of a feature on the part to measure, the result will be inaccurate. Moreover, the point clouds generated if the machine is used to measure absolute point coordinates instead of relative distances may not be accurate since the probe geometry has to be taken into account. CMM manufacturers provide a compensation mode to correct for the probe geometry (Figure 1 [4]).

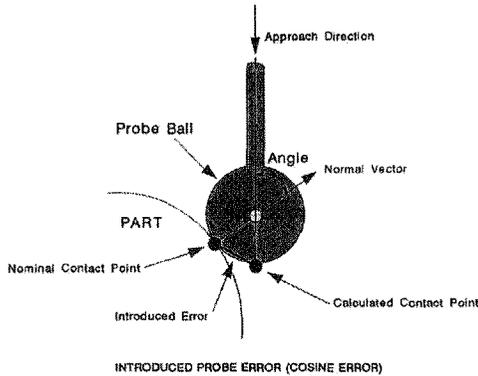


**Figure 1: Uncompensated and Compensated Measurements**

For the reverse engineering application, absolute data measure is required, and the compensation factor is essential. It is however very difficult to acquire correctly the coordinates of the points touched by the probe because the standard procedure is to use a standard offsetting method with the ball radius along the probing direction. One can see that the results are not always correct and depend on the normal vector angle (Figure 2 [4]). Some errors are not negligible [5](Table 1[4]).

Two other probe parameters are critical. The first one is the rigidity of the stylus. The smaller the probe, the more the stylus is subject to bending. This issue is very important if the fast speed acquisition mode is chosen. The second parameter is related to the geometry of the stylus. Some geometries cannot be used to acquire data if the part geometry interferes

with a feature on the part. Thus, the data acquisition aspect requires experience and know-how in order to generate usable data.



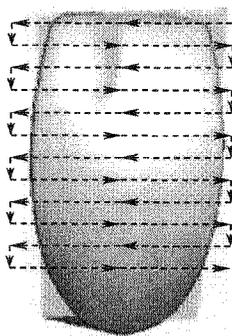
**Figure 2: Error from the Compensation Procedure**

Probe diameter (mm)	0.50	1.00	2.00	3.00	4.00	6.00
<b>Angle to normal (degs)</b>	Magnitude of error introduced by not probing normal to surface					
1.00	0.0000	0.0001	0.0002	0.0002	0.0003	0.0005
2.00	0.0002	0.0003	0.0006	0.0009	0.0012	0.0018
3.00	0.0003	0.0007	0.0014	0.0021	0.0027	0.0041
4.00	0.0006	0.0012	0.0024	0.0037	0.0049	0.0073
5.00	0.0010	0.0019	0.0038	0.0057	0.0076	0.0115
7.50	0.0022	0.0043	0.0086	0.0129	0.0173	0.0259
10.0	0.0039	0.0077	0.0154	0.0231	0.0309	0.0463
15.0	0.0088	0.0176	0.0353	0.0529	0.0709	0.1058
20.0	0.0160	0.0321	0.0642	0.0963	0.1284	0.1925

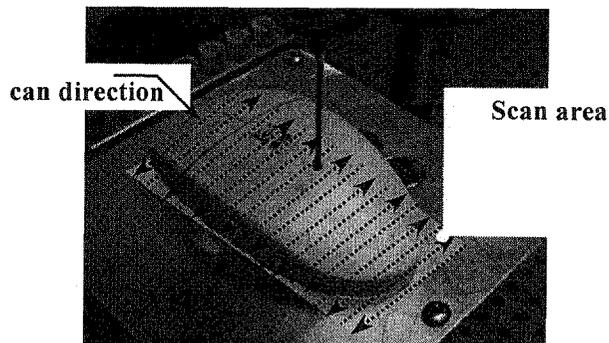
**Table 1: Values of the Acquisition Error Depending on the Probe Diameter**

### Measurement procedure

To understand the tessellation algorithm, it is necessary to understand how the CMM proceeds to obtain the data point cloud.



**Figure 3: The way of scanning**



**Figure 4: Definition of the parameters for scanning**

The MMIV software scans the part through a point grid (Figure 3) and results in a regular point data distribution. The number of points for each line is constant. The process of scanning requires the definition of the limit of the grid and of the number of points per line (Figure 4). The main problem is to define the best measurement procedure to capture all the details without unduly increasing the scan time.

### STL file generation

After the digitization of the part is completed, the STL file can be generated from the data point clouds. The STL format, introduced by 3D-Systems©, represents a geometry with a series of triangles. Each triangle has exactly three adjacent triangles (one along each edge) and is assembled in a surface defining a closed volume.

Three main topics are developed in the following paragraphs to explain how the STL file is created from a list of points representing one surface. The resulting algorithm is summarized in the flowchart of Figure 5.

### Tessellation

The process of tessellation needs to follow the appropriate rules to generate properly triangular surfaces conform to the STL format.

The tessellation algorithm developed in this paper uses the structure of the CMM data. Indeed, sorting of the data points depends on the grid, and the tessellation works on two lines of data points at a time. (Figure 6). The tessellation algorithm described in the next paragraph for two data lines, is illustrated in the flowchart shown in Figure 7.

In Figure 6, tessellation is performed from the left to the right. The two points located on the left side (A, 1) are initially considered, each of these points being on their respective line. Triangle formation depends on the choice of a third point, which can either be B or 2, both of these points being located immediately to the right of the two points considered. The shortest diagonal dimension (A2 or 1B) will determine the point to be selected. Once all three points are determined, the triangle can be built. The selection sequence starts over and the two initial points to consider are located on different lines and on the outer right side of the last generated triangle (B, 1).

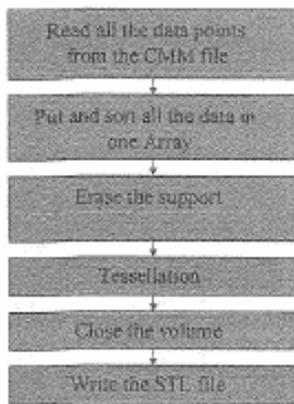


Figure 5: Main algorithm

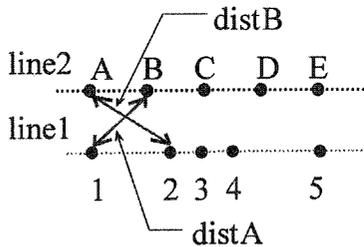


Figure 6: Tessellation with 2 Data Lines

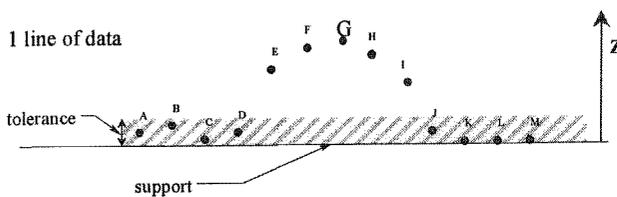


Figure 8: Determining where the Support is.

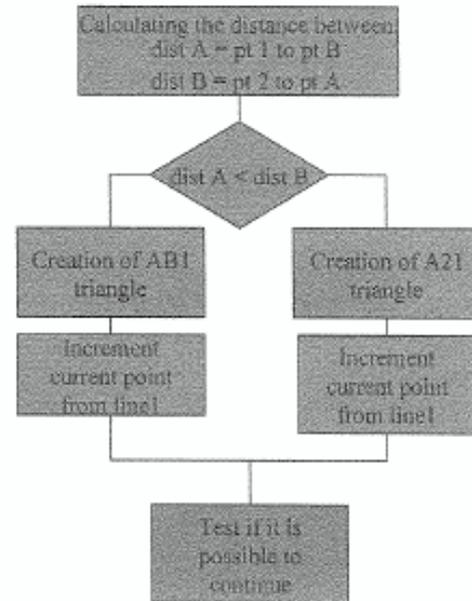


Figure 7: Tessellation Algorithm

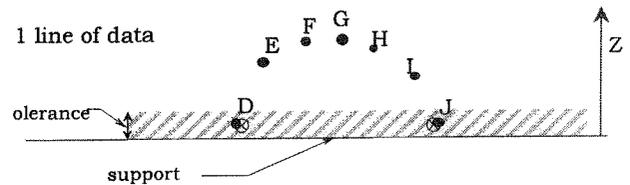


Figure 9: Suppression of the Support

### Erasing the ground

When using the CMM, the supporting platform of the part needs to be digitized in order to capture all the required details. Thus, the proposed algorithm consists of defining the points of the part without the support points. Since the support is flat and at the bottom of the part, a custom height is set to delete all the points included within this tolerance, starting from the lowest point (Figure 8). The major constraint is that two points must be kept to represent the ground during the tessellation (Figure 9). The height of the part will not be accurate if this process is not used. Furthermore, the program deletes the complete line of points if they all belong to the support. Therefore, the tessellation is performed only on the points that belong to the part.

### Closure of the volume

The result obtained so far represents a surface and not a volume, and the last operation must close the surface to correctly define the STL file. For a convex part, “three” surfaces are ignored at the end of the tessellation since:

- The CMM only scans the top surface (no data about the bottom of the part)
- The tessellation proceeds only slice per slice in the YZ-plane

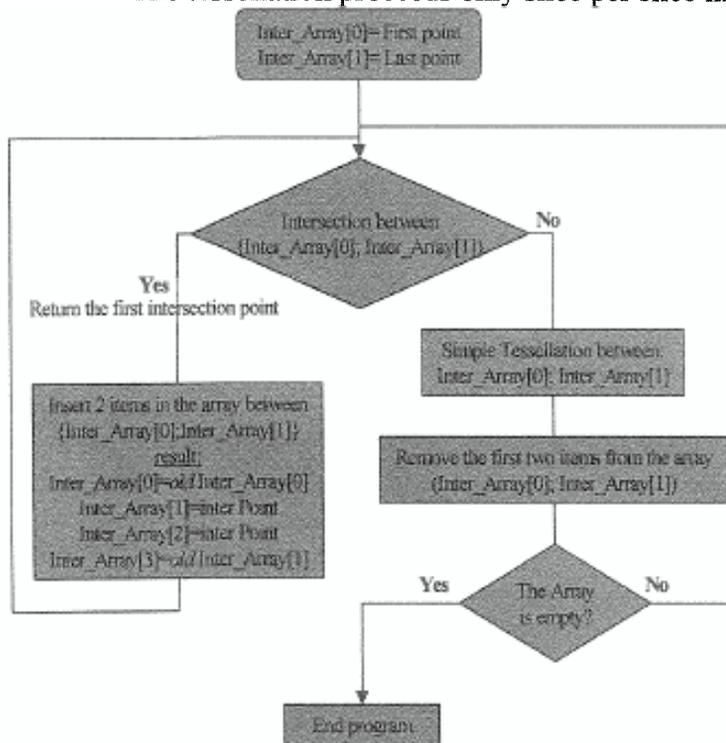


Figure 10: Edges Tessellation Flowchart

Thus, the under-side and two sides (right and left in this case) are not digitized, and without these surfaces, the volume is not closed. Therefore, it is necessary to implement another algorithm, to generate the missing surfaces.

The tessellation for the under surface is simple and proceeds like the previously described procedure (Figure 11). Edge tessellation is more complex. The form of the data points does not follow a set pattern (Figure 12). The algorithm is explained in the flowchart shown in Figure 10.

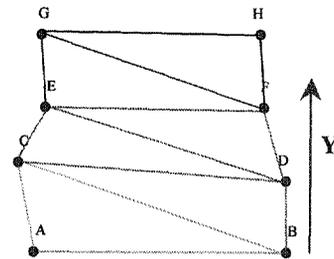


Figure 11: Back Tessellation

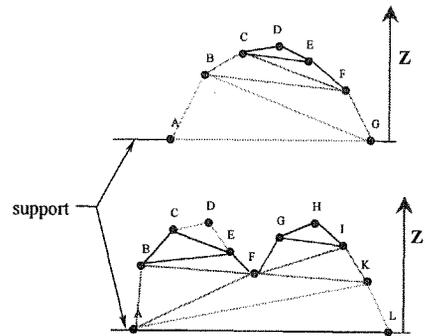


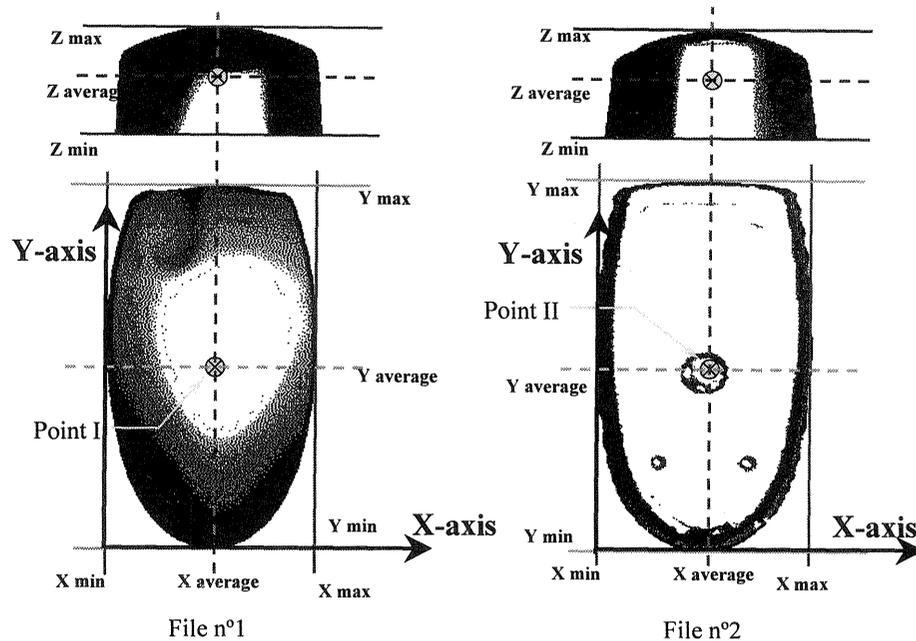
Figure 12: Edges Tessellation

## Multiple Faces

Having scanned the required surfaces, a method is needed to merge them.

### *Common space*

The main problem is to find a common space. Each surface acquisition has its own referential. Therefore, merging two surfaces without common referential is very difficult. The idea is to use certain scanning parameters. Indeed, defining the axis and origin, which are needed before the digitization, can be used to correctly orient the surfaces. Upon definition of a referential with the CMM, all rotational degrees of freedom are locked.



**Figure 13: Operations to Obtain the Common Space**

In fact, these references determine unique X and Y axes for all digitization. Thus, if the definition is correct, surfaces are well oriented with respect to each other but not yet centered.

The next step is to translate the surface to a same origin. To find a common point (Point I and Point II), it is assumed that the overall dimensions of the part remain unchanged, and the middle of the largest dimensions in X and Y provide the reference point needed (Figure 13).

### *Merge method*

The merging operation depends on the angle of the added surface. The most difficult part of this operation is to correctly sort the data points. In fact, the tessellation algorithm presented in section "Tessellation" which sorts the data points, is also used here.

The simplest method is to merge the top and under surfaces. The junction and stitching is indeed straightforward since both surfaces have almost the same area and only a few common points. However, the edge surfaces represent a new view of the part, which makes it difficult to identify common points.

For the back surface ( $180^\circ$  rotation), the algorithm matches the points using the Z-axis level (Figure 14). For all other edges, the distances between points from both surfaces (Figure 15) are calculated to determine the order of the points.

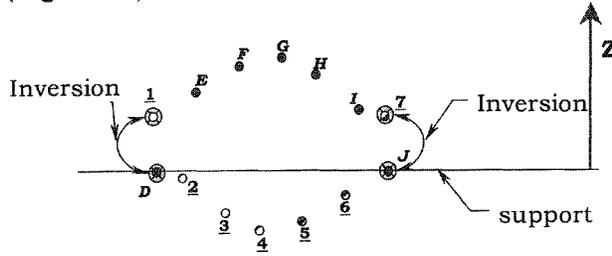


Figure 14: Merging with  $180^\circ$  Surface

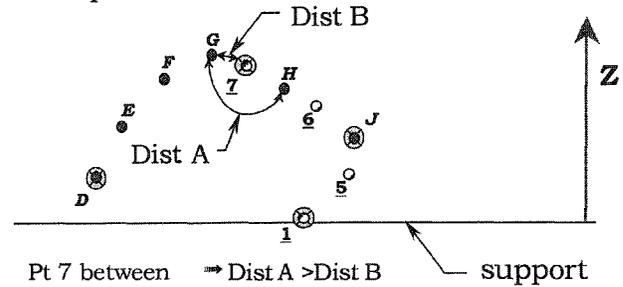


Figure 15: Merging with  $90^\circ$  and  $-90^\circ$  Surfaces

### ***Problems:***

This algorithm has some limitations. During the merging operation, all the points from the direct part (without the support) are used even if they are inaccurate (point on the support closest to the part and retained by the algorithm). Therefore, some points may be eliminated during the stitching.

The other limitations come from the tessellation, which works only with two lines of data points at a time. Due to this characteristic, the merging of two surfaces can only occur when the same number of lines and thus the same scanning grid for both surfaces are used. This aspect prevents optimizing the scanning time that can result from employing different grid sizes for surfaces with simple geometries.

### **Improvement of the STL file**

The research presented utilizes a CMM machine to digitize a part. In order not to miss any geometrical detail, a large number of measurement points are normally acquired. If all the points are used in the STL file generation, the size of this file can easily become too large. In fact, keeping lots of data points in nearly planar regions is redundant. Removal of data points in these regions would not affect the accuracy for rapid prototyping machines [6]. There are different methods to solve this problem. The first consists of building a decimation algorithm meant to work directly on the STL file. The second is based on decreasing the number of points before the tessellation is performed. This latter alternative being the simplest and the fastest, was retained and is detailed below.

#### ***Deleting redundant points***

To determine which points are essential, the program uses a user input acceptable tolerance height  $h$  between two vectors ( $EF$  and  $EG$  in Figure 16). The variable  $h$  is obtained with the calculated angle  $\beta$ . If the calculated  $h$  is less than the prescribed one, the middle point is eliminated.

This method has some limitations. In fact the decimation works only in one direction. Also, the reduction of points is not optimal, and the neighboring lines are not taken into consideration. Nevertheless, with this method, the tessellation algorithm can still be used and the processing remains fast.

It was observed that on average about 80% of the points were eliminated through linear decimation thus contributing to a significant reduction of the STL file size.



- The decimation method can help to achieve a better accuracy. The decimation reduces the number of points and improves the flat surfaces. Moreover, decimation improves the junction of two surfaces. For example, with too many points, it may be difficult to determine the corners of the part. Using decimation, the corners appear like chamfers or sharp edges.

The problem of good accuracy depends on the user's intention. In fact, the choice of parameters should depend on the surface characteristics. For cylindrical surfaces, it is better to choose a small grid size and a small decimation tolerance. For a plane, the grid needs to be fine only if the contour of the part is important.

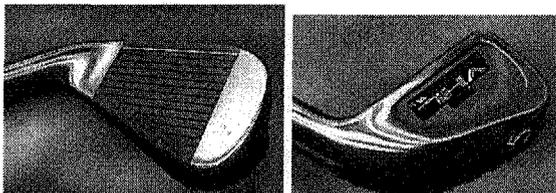
An improvement in accuracy (specifically the contour) can be achieved with the merge function. After the merging operation, only points from the part are kept, resulting in a more accurate STL file. Unfortunately, to use this function, some special supports need to be built to maintain the part in position. The example of the golf club head below illustrates the merging function's performance.

The other important observation is that the CMM is not aware of the space spanned by the part for digitization. The user needs to set grid size and limit points to allow the machine to reach the part's edges. Therefore, the contour and the quality of the copy part depend on these parameters. In the cases presented, the grid size was arbitrarily set to keep exactly the same number of points. Therefore, sometimes the results were better with a larger grid size. For the test part, the accuracy obtained was within 0.2 millimeters.

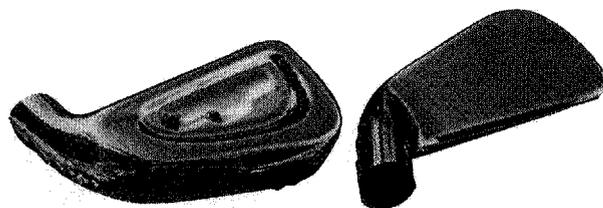
### Results with the merge method

For a second example, a golf club head was chosen as a model. Since both sides are relevant, this part is a good application for the merge function.

One scan was applied to each surface. The creation of an assembly was necessary to digitize accurately the opposing surfaces. Furthermore, the setting of the digitization software had to be the same, otherwise the merging would fail. Several tests were performed with different grid sizes and different tips. The best results were obtained with a 70 \* 70 points grid and a 3 mm tip. The entire digitizing process took 12 hours per surface. At the end of the process, the part obtained and shown in Figures 17 and 18, can be considered a good copy of the original part. Nevertheless, some problems are present where both surfaces meet each other. Furthermore, due to the height of the part, the digitizing process used a long tip to enable data capture. As a consequence, some details of the groove were lost.



**Figure 17: Original part**



**Figure 18: Copy of the golf club**

### Conclusion

The work presented in this report is focused on the development of a methodology and program, which allow direct STL files generation from real parts, using CMM for data acquisition.

Explanations are provided on how the software works and which parameters the user needs to set to build a good copy. The research also proposes a method to merge different surfaces. This new approach shows that it is possible to use a CMM to completely digitize a part, although the probe may not be able to acquire all the different surfaces in one sequence.

A decimation algorithm is described to simplify and decrease the size of STL files. Several tests show that it helps to obtain more accurate results.

Nevertheless, reverse engineering smooth surfaces like those of the golf club head is more difficult than simple geometries such as those of the test-part. Furthermore, it is almost impossible with this technique to represent corners or perfect planes with only the points acquired by the CMM. Although the results proved to be satisfactory, inaccuracies between original and computerized models were mainly due to a malfunction of the compensation program of the CMM.

### **Future work**

Future work should be aimed at improving and obtaining a complete reverse engineering procedure using the CMM as a primary data acquisition routine. First of all, a more detailed study needs to be achieved to demonstrate the accuracy of the results when using the compensation program. As mentioned in this report, this function did not work properly in our software version.

The program does not consider hollow areas on the original part. Also, parts with holes are not correctly processed. Furthermore, the decimation program works only in the XZ direction. Further reduction in size of the STL file could be obtained if the decimation was also performed in the YZ direction.

During the tessellation, a method could be proposed to add some points when necessary. For example, if the angle displayed in Figure 16 is too important, a new point resulting from the intersection of lines EF and GH created with the points located before and after the points F and G should be considered. The case of the corners is an example.

These proposed improvements along with the continuous growing interest in reverse engineering would then make it possible and very interesting to propose this new concept to companies without significant financial investments.

### **References**

- [1] Harry Hutchinson, "*Relief for weary bones*", Mechanical Engineering; Vol. 121, No. 6, pp 54-57, June 1999
- [2] Kwan H. Lee and H. Woo; "*Use of reverse engineering method for rapid product development*"; Computers Ind. Engineering Vol. 35 pp. 21-24 1998
- [3] William B. Thompson, Jonathan C. Owen, H. James de St. Germain, Stevan R. Stark and Thomas C. Handerson; "*Feature-Based Reverse Engineering of Mechanical Parts*"; IEEE Transactions on Robotics and Automation, Vol. 15, pp 57-66 1999
- [4] "*MMIV Software Reference, Manual*"; Brown & Sharpe Mfg. Co.; Training Brief #15-2, 1994

[5] C. Menq and F. L. Chen; "*Curve and Surface Approximation from CMM Measurement Data*"; Computer Ind. Engineering; Vol. 30, No. 2, pp 211-225, 1996

[6] Y. H. Chen, C.T. NG and Y.Z. Wang; "*Data Reduction in Integrated Reverse Engineering and Rapid*"; Computer Integrated Manufacturing, Vol. 12, No 2, pp 97-103, 1999

