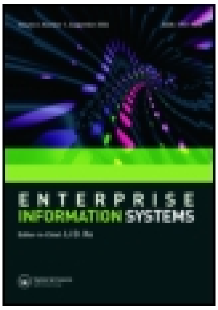
# Sensing and responding to the changes of geometric surfaces in flexible manufacturing and assembly

**Article** *in* Enterprise Information Systems · May 2012

DOI: 10.1080/17517575.2012.654826



Z. M. Bi
Indiana University-Purdue Univer…




Enterprise Information Systems
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/teis20

# Sensing and responding to the changes of geometric surfaces in flexible manufacturing and assembly

Z.M. Bi [a] & Bongsu Kang [a]
[a] Department of Engineering, Indiana University Purdue University Fort Wayne, Fort Wayne, IN, 46805, USA

Available online: 06 Feb 2012

*Enterprise Information Systems*
2012, 1–21, i*First* article



# Sensing and responding to the changes of geometric surfaces in flexible manufacturing and assembly

Z.M. Bi* and Bongsu Kang

*Department of Engineering, Indiana University Purdue University Fort Wayne, Fort Wayne, IN 46805, USA*

(*Received 4 February 2011; final version received 3 January 2012*)

A flexible manufacturing system is capable of dealing with changes and uncertainties occurring in a manufacturing environment. A rigid automation system adapts changes through reconfiguration or manual modification of its control programmes. In contrast, an advanced system responds to the changes autonomously or with a minimal manual intervention. One critical technology for a flexible manufacturing system is to detect or predict the changes and uncertainties and take them into account for the control of the manufacturing system. In this paper, an intelligent manufacturing system is proposed to accommodate product geometric variants autonomously by integrating a 3D vision system into manufacturing processes. The proposed system is capable of acquiring vision sensor data, detecting the geometric changes, and modifying the control programmes of the machine in response to the changes. In this innovative work, a new method for converting the raw vision sensor data into a surface model of the product is presented. A new surface reconstruction algorithm is proposed to generate the surface model directly from the raw 3D point-cloud data of a product. In addition, an advanced programming algorithm that uses acquired surface models to automatically update robot programmes for a new task with minimal manual intervention is presented.

**Keywords:** enterprise information system; flexible manufacturing system; automation; vision-based detection; surface reconstruction; marching algorithms; ITK/VTK

## 1. Introduction

An *enterprise information system* aims at providing high-quality service and dealing with large volume of data; it is a hierarchical technology platform that enables users to integrate and coordinate their business processes. An enterprise information system consists of sub-systems at different levels. This paper focuses on the information integration at the machine. The main purpose is to enhance the level of automation and flexibility for flexible manufacturing systems (FMSs), which are required to deal with the changes in their application environments.

Regardless of the level of an enterprise information system, its intelligence can be partially measured by its capability of dealing with the changes and uncertainties in a dynamic business environment; there are many causes of these changes and

*Corresponding author. Email: biz@ipfw.edu

ISSN 1751-7575 print/ISSN 1751-7583 online

http://dx.doi.org/10.1080/17517575.2012.654826
http://www.tandfonline.com

uncertainties such as (i) changeable variants of product, (ii) the fluctuated product volumes, (iii) changes of order from customers, and (iv) unpredicted events occurred to the production line (Li 2000a, 2000b, Xu *et al*. 2005, 2007, Li 2007, Li *et al*. 2008, Zhang *et al*. 2011). Changes can only be tackled by changes, i.e. making a system flexible or reconfigurable is the most appropriate solution to introduce extra variables into a system (Bi and Zhang 2001). Intensive researches have been conducted to develop various FMSs, in particular, on agile manufacturing and reconfigurable manufacturing (Mehrabi *et al*. 2000, ElMaraghy 2005, Yin and Xie 2011). One may note that the need of information integration for changes and uncertainties is not unique to manufacturing applications; any enterprise information system has a similar need (Yin *et al*. 2012). Comprehensive summaries on the development of intelligent systems with flexibility and reconfigurability were provided by Bi *et al*. (2007, 2008).

Many FMSs are rigid automation systems, which have very limited flexibility to respond to the changes and uncertainties during manufacturing/assembling processes. This limitation is caused by insufficient information about the changes and the incapability of changing the control programmes of the machine for new tasks. Studies on the development of advanced sensing systems to acquire data from a manufacturing environment and intelligent programming systems to generate control programmes for new tasks are rare. Therefore, innovations are very demanding on the following aspects: (i) to acquire and interpret data related to changes and uncertainties and (ii) to respond to the identified changes and modify the control programmes of flexible machines automatically. The present work is a recent achievement of the author built up on earlier effort in developing an automated robotic coating system (Bi and Lang 2007a, 2007b). The present work is focused on the development of a sub-system for data acquisition and processing. This sub-system is capable of acquiring the 3D point-clouds data of product surfaces and interpreting them for the representation of geometric surfaces; the robots can be then programmed by an intelligent system to accomplish the corresponding operation on a new product. Note that the present algorithm can be generalised and applied to other surface treatments such as polishing, geographic surveying, mowing, and agricultural field operations.

In this section, the motivation of the present study is addressed. First, the purpose of surface reconstruction for different applications is discussed. Secondly, existing algorithms for surface reconstruction are overviewed. Thirdly, the limitations of the current available technologies for surface reconstruction are summarised. The last section is concluded by a summary of the present work.

## 2. Surface reconstruction for manufacturing and assembly

With the advance of information technology, computers became essential tools for most of the daily operations, in particular, for manufacturing automation. To this end, representing an object by a digital model for computers is usually the first step to use *computer aided design* (CAD) and *computer aided manufacturing* (CAM). However, a digital model is not always available to a computer system in many situations: (i) objects in nature, such as regional landscapes, have existed for millions of years before computers came into our lives; (ii) growing objects, such as bones in human bodies, growing up naturally without a design sketch and (iii) derived objects. Although objects in manufacturing are made based on an artificial design in general, some changes are made in their actual production processes, and there are discrepancies between as-designed models and as-built objects. When the computer geometric model of an object is concerned, the modelling process from the acquired data to the surface model is called *surface reconstruction*. More correctly, surface reconstruction was defined by Hoppe (1994) as *given the partial information of an unknown surface, construct, to the extent possible, a compact representation of the surface*.

Surface reconstruction techniques have been successfully applied in medical imaging, cartography, computer arts, and visualisation of cultural and scientific heritage. Many researchers have published their works on different algorithms for surface reconstruction. For example, Mari and Mari (2004) developed a MATLAB toolbox to analyse 3D images and reconstruct the surface model from an unorganised set of points. Rocchini *et al.* (2001) presented a suite of the tools to visualise cultural and scientific heritage based on the acquired point-clouds data. Shih and Wang (2004) and Rebolj *et al.* (2008) introduced the system used to monitor the construction process by comparing as-scheduled and as-built models. In manufacturing industry, most of the studies were focused on reverse engineering and a surface model of object is usually generated from its digital solid model (Várady *et al.* 1997, Thompson *et al.* 1999, Fisher 2002, Pernkopf 2005). For surface reconstruction, Song *et al.* (2005) worked on the automatic surface reconstruction using a fitting approach with an assumption that the scene was an aggregate of the objects from simple parametric surfaces such as flat planes, cylinders, and spheres. The point clouds can be matched and used to determine the parameters of simple objects. As a case study, the plant model was created from the point clouds which were acquired by laser sensors. Biegebauer *et al.* (2002) and Vincze *et al.* (2002) developed a sensor-based robotic painting system. The geometrics of parts are acquired by the vision sensor and the programmes are generated by computer for painting operations. The studies were similar to the previous work (Bi and Lang 2007a); however, their surface reconstruction system was confined to small parts, and the task of data process was performed separately by commercial software tools. Their system could be seriously challenged to be applied for large and complex products in a short cycle time. These challenges will be discussed in Section 4.

Automated tasks of surface treatment are defined based on given surface models. One critical requirement of surface reconstruction for manufacturing automation is the reconstructing time for an acceptable surface model. It must be shorter than the cycle time of the corresponding surface treatment in the production line. In addition, surface reconstruction must be automated so that no special expertise is required from an engineer who controls and monitors the surface treatment. In Section 3, some popular algorithms of surface reconstruction are reviewed and their limitations to meet these critical requirements are discussed.

## 3. Algorithms for surface reconstruction

Surface reconstruction from the point clouds is fundamental to numerous engineering applications. This generic problem has attracted many researchers. A few of the comprehensive literatures on surface reconstruction were given (Azernikov *et al.* 2003, Gois *et al.* 2004, Schall and Samozino 2005). We summarised these reviews based on Azernikov's classification: *computational geometry approaches* and *computer graphics approaches*.

The approaches in the first category focus on the piecewise-linear interpolation of unorganised points (Allgower and Schmidt 1985), and define a surface as a carefully chosen sub-set of *Delaunay Triangulation* in a Cartesian coordinate. Delaunay triangulation was discussed in detail by Cazals *et al*. (2004). Amenta *et al*. (2001) used a medial axis transform for the approximation of a surface. The *medial axis transform* was a representation of an object as an infinite union of balls; the surface approximation was called as *power crust*. Bernardini *et al*. (1999) developed the ball-pivoting algorithm for surface reconstruction based on a simple principle: three points form a triangle if a ball of a user-specified radius touches them without containing any other point. Therefore, beginning with a seed triangle, the ball pivots around an edge until it touches another point to form the next triangle on surface. Abdel-Wahab *et al*. (2005) conducted a comparison of some computer-geometry based approaches including the crust, power crust, tight cocone, and ball pivoting algorithm; the criteria they used were quality, memory usages, and time of reconstruction. They concluded that the crust and power crust algorithms showed a balanced trade-off between the execution time and the memory usage. The ball pivoting algorithm exhibited the minimum execution time and memory usage, which was followed by the tight cocone algorithm. The experiments showed that applying any of the four algorithms on a non-uniformly distributed cloud may create poor quality surface. In general, the computational cost of the algorithms under this category is determined by the generation of Delaunay triangulation. They can be applied only when the point clouds do not include noise, which is impractical to the raw data from 3D laser scanners.

The approaches in the second category target on the best approximation to an imagined surface for the whole point clouds. Individual points are not necessarily confined to this surface and a discrepancy within a specified tolerance is acceptable. Curless and Levoy (1996) proposed a volumetric approach which exploited the fact that the point clouds were a collection of laser range images. Unfortunately, it is restricted to devices where the projection plane is known. The fundamentals of the approaches in this category were established by Hoppe *et al*. (1992) and Hoppe (1994). The approaches were based on the idea of determining the zero set of an estimated signed distance function. They are capable of inferring the topological type of the surface automatically, including the presence of boundary curves. The methods of defining a signed distance function were also adopted by Neugebauer and Klein (1997). Freedman (2004) proposed an incremental technique for surface reconstruction. The algorithm does not require the surface's embedding space; the dimension of the embedding space may vary arbitrarily without substantially affecting the complexity of the algorithm. Zhao *et al*. (2001) introduced an algorithm based on variation and partial differential equation methods.

## 4. Limitations of existing algorithms

Enhancement of flexibility for a manufacturing system is critical to deal with changes and uncertainties in a dynamic manufacturing environment. The first step towards this goal is to identify the changes and to represent them in computer models. An intelligent programming system then takes into account these changes and creates updated control programmes for new tasks online. However, two of the key obstacles preventing the use of digitalised technologies for wide applications in manufacturing industry are the strict requirement of a short processing time and the difficulty of automation.

*Processing time*. Different from other successful applications of surface reconstruction mentioned in Section 2, an application in manufacturing usually has a strict requirement of a short cycle time of task. It is particularly true when the tasks in a product line are considered; the whole production system should not be blocked by a single process due to a longer processing time for data acquisition. If a task, such as a coating operation in final assembly line, involves a surface reconstruction, the time to reconstruct a surface model must be less than the cycle time that the system spends on every product. However, most existing algorithms are inefficient to be applied in actual production systems.

*Difficulty of automation*. Automated machines, such as robots, have to be programmed such that the changes of tasks can be accommodated in the corresponding operation. In other words, a software system is needed to identify the changes and take them into consideration automatically. Existing technologies for FMSs still need considerable manual interventions to generate control programmes for new tasks, e.g. coating on modified surfaces.

Although not discussed, efforts to overcome other limitations, not specifically for manufacturing applications, should be also recognised. For example, the size of a data set is a primary concern for surface reconstruction and visualisation, in particular for web-based applications. Engel *et al.* (1999) introduced an adaptive and hierarchical concept to minimise the number of vertices that have to be reconstructed, transmitted, and rendered, and that the resulted system was able to directly generate the stripped surface representation in a web-based application.

## 5. Organisation of the paper

The main goal of the present work is to find a solution to detect geometric changes of a product and reconfigure the control programme of a FMS for new tasks effectively. Industrial robots are considered as examples of FMSs, and their flexibility is achieved by modifying their control programmes. The remainder of the paper is organised as follows. In Section 5.1, the relations between the technical approaches of a FMS and changes in a manufacturing environment are discussed and the scope of the present work is defined. System architecture to respond to real-time changes in a production line is presented. In Section 5.2, technologies for sensing and interpreting the changes are discussed; a new algorithm for surface reconstruction is presented. In Section 5.3, a system model is developed to use the detected changes for closed-loop control of the robotic system. In Section 5.4, summary, conclusions, and future research directions are presented.

### *5.1. Flexibility vs. changes*

In terms of its strategic objective to deal with changes and uncertainties occurring in a manufacturing environment, an advanced manufacturing system is indeed a combination of a *FMS* and *modular manufacturing system* (MMS). FMS is an integrated system and achieves its flexibility via adjusting some internal parameters by software, while MMS uses modular architecture and achieves its flexibility by various system configurations assembled from a set of modules. FMS and MMS have different levels of reconfigurations.

As shown in Figure 1, in order to maximise system flexibility, an advanced manufacturing system includes some key elements from both FMS and MMS, i.e. it adopts modularised architecture to allow the system to be reconfigured at the system level, and it consists of some encapsulated modules with the flexibility for modification of control programmes and accommodating new tasks. Correspondingly, an advanced manufacturing system deals with two classes of changes: *short-term* and *long-term* changes. Short-term changes, such as a change of robot path, can be tackled by reconfiguring control programmes (software reconfiguraion); control programmes are determined when the requirements of new tasks are known. Long-term changes, such as a change of business scope from automobiles to aerospace, have to be dealt with radical changes of system configurations (hardware reconfiguraion). Long-term changes are beyond the reconfigurability of a single hardware configuration. Long-term changes are unanticipated in a short term, but they have to be predicated when system architecture of an advanced manufacturing system is developed. The scope of this paper is confined to the short-term changes which can be met by software reconfiguration.

As shown in Figure 2, from the point of view of the hardware system, an advanced manufacturing system consists of two classes: (i) the sensor-based detecting system to identify and interpret the changes and (ii) the machining system to perform actual manufacturing operations. From the point of view of the software system, reconfigurable machines need to be programmed. Since each programme can only deal with a specific set of task requirements and the occurred changes will affect the task requirements. A comparison and correction tool is required to capture and export the changes so that the programmes for reconfigurable machining systems can be modified to accommodate the changes.

Notice that the sensing system is intentionally separated from the rest of the FMS shown in Figure 2. It is a distinguished feature of FMS from a dedicated and rigidly

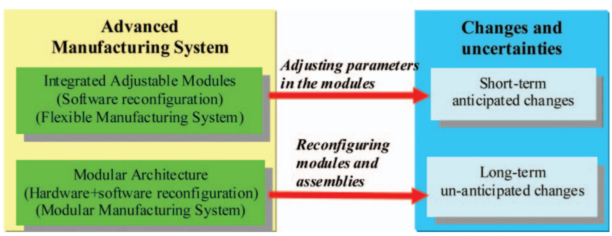



Figure 1. Advanced manufacturing system vs. changes.

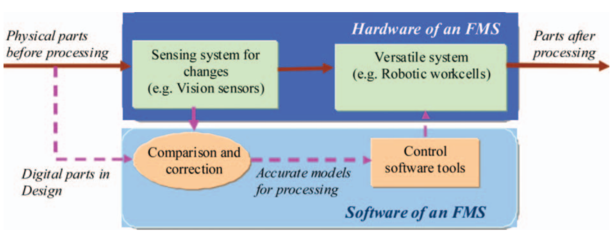



Figure 2. A generic FMS model.

automated manufacturing system. Since FMS is designed to meet the changes, and the changes have to be defined first during the system operation. Figure 2 is a generic model of FMS. In an extreme case, one may remove the sensing system if task variants can be completely defined by using available CAD models and the parts can be precisely positioned when they are transferred into the system.

Hereafter, our concentration will be moved to two basic tasks: (i) sensing and interpreting the changes and (ii) responding to the changes by modifying control programmes of the machine. These tasks will be discussed in the following two sections. To facilitate the discussions, geometric changes are concerned and the applications of FMS are limited to the surface treatments where correct geometric models are essential to create control programmes.

### 5.2. *Sensing geometric changes*

In surface treatments such as painting, coating, polishing, or grid-searching, the machine movements are defined based on the surface models of an object. In manufacturing, the task requirements will be changed frequently if a production line has mixed type of products, and/or the product itself is an assembly of many small parts, and different parts in the same family can be exchanged in assembling. The surface models can be retrieved from CAD models. However, when CAD models are not available or they are not accurate or updated for machine programmes, sensing the geometric changes of the product will be the first challenge to the implementation of software reconfiguration.

#### 5.2.1. *Tasks in sensing and interpreting changes*

Stereo cameras or laser sensors can be applied to capture 3D range data of a surface. When raw surface data are acquired, they need to be processed and interpreted for the changes of task requirements. The interpretation of raw surface data for a certain computer model is very complicated process. As shown in Figure 3, the following typical tasks are involved in this process.

*Data filtering*. The raw data include noise, distorted, and invalid data caused by the hardware system and environment. The raw data have to be filtered to remove unwanted and noise data. Commercially available vision systems are mostly

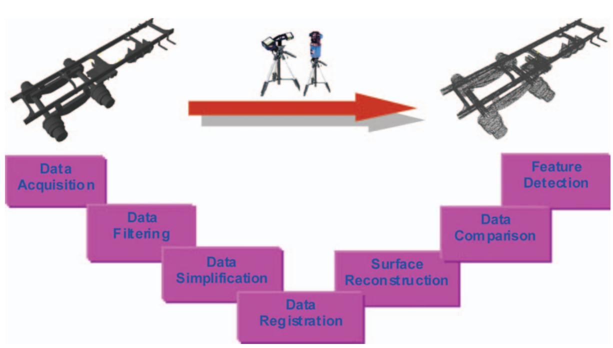



Figure 3. Tasks in sensing and interpreting geometric changes.

accompanied by their own tools to filter raw data. Those systems do require users to specify the sensing ranges where an interested object is placed.

*Data simplification and smoothing*. Three-dimensional point-clouds data usually require a large amount of computer memory to store and transform the data, and thus a long processing time. Therefore, it is necessary to simplify the original raw data to lower the memory requirement and accelerate data processing. In addition, the simplification and smoothing of surface data can also benefit other activities in off-line programming of the machine such as collision detection, visibility testing, shape recognition, and visualisation in simulation.

*Data registration and integration*. A vision sensor can only capture the part of the surface which faces the sensor (Bi and Wang 2010). In acquiring the data for a complete surface, data from multiple views are needed for surface reconstruction. The data from different views are then fused together to define the surface model. Moreover, data registration is used to determine the transformation of the data from two different views so that the data can be integrated under the same coordinate system. Integration is the process of creating a single surface representation from two or more range images. When the object is large, multiple sensors may be needed or a sensor needs to be equipped with actuators to observe the object from different views. In both cases, the information of the relative position and orientation between the sensors and the object must be known for data registration and integration. As a result of data registration, all the point-cloud data for the same object are merged together with the same global coordinate system.

*Surface reconstruction*. Based on the integrated raw point-cloud data acquired from the surfaces of an object, surface reconstruction is to generate a surface model that approximates the actual surfaces of the object from which the raw point-cloud data are acquired. If the designed surface model of the object is available at the same time from an existing database, reconstructed surfaces can be also used to compare the original design to identify changes and generate machine programmes later for automation.

*Feature detection*. Fasteners, extrusions, bosses, and holes on a surface can be dealt with features. Recognition of these features is usually very important for manufacturing processes. Parameters of such features include size, position, and contour measurement which can be determined by post-processing such as edge detection. Feature detection is used to identify the features with certain types of properties or justify if the acquired data correspond to a specific feature.

*Data comparison*. A reference virtual model should be available in data comparison. Data comparison is used to calculate the deviation or differences of the actual physical model from the reference virtual model. Comparison of the two models is required for some operations such as alignment and inspection. In actual applications, data comparison can be merged with feature recognition. For example, (i) point-cloud data and a designed CAD model can be compared to identify geometric features such as plane, cylinder, circle, and sphere; (ii) in generating control programmes of machine, as-designed and as-built models are compared so that the deviation (average error), tolerance, and distribution can be evaluated.

### *5.2.2. New algorithm for surface reconstruction*

Surface reconstruction is to define a surface $U$ from a set of unorganised data points $P = \{p_1, p_2, \ldots, p_n\}$, which are assumed to be on or close to this surface. Hoppe's algorithm (Hope *et al.* 1992, Hoppe 1994) is used as the benchmark algorithm. In this algorithm, the key procedure is to estimate the scalar distance of an arbitrary point $p \in R^3$ to the surface $U$. The distance is signed and defined as $d_U(p) = s(p) \cdot d(p, U)$, where $s(p)$ denotes the side of the surface the point $p$ lies. Knowing the signed distance function $d_U$ is equivalent to knowing the surface $U$. Therefore, the algorithm is to estimate $d_U$ from the data points and then extract an approximation of its zero set. The estimated scalar distance $\tilde{d}_U$ is associated with an oriented plane with each of the data points. At each data point $p_i$, the oriented plane is the tangent plane $T_p(x_i)$ which is a linear approximation to the surface. This plane is represented by a point $o_i$ and its unit normal vector $\hat{n}_i$. The scalar distance $\tilde{d}_U$ of an arbitrary point $p \in R^3$ to $T_p(x_i)$ is defined by $d_i(p) = (p - o_i)\hat{n}_i$.

The point on the plane and its normal vector for $T_p(x_i)$ are determined by gathering together the group of points of $P$ within the distance $\rho + \delta$ of $p_i$ (where are $\rho$ and $\delta$ are the parameters estimating the sampling density and noise, respectively); this set is denoted by $Nbhd(x_i)$ and is called the neighbourhood of $p_i$. The centre $o_i$ is taken to be the centroid of $Nbhd\,(x_i)$, and the normal $\hat{n}_i$ is determined using the principal component analysis. After $\tilde{d}_U$ is defined for all points in the area of interest, a contouring algorithm can be applied to extract an approximation to $Z(\tilde{d}_U)$ in the form of a mesh. However, a few limitations of the Hoppe's algorithm for manufacturing applications are as follows:

- The underlying surface has to be a manifold. However, the acquired surfaces in many applications fail to meet this assumption.
- The satisfaction of the linearisation of the underlying surface depends on the parameters of $\rho$ and $\delta$. They are specified based on the density and noise of the samples. However, the density and noise vary significantly with the procedure of data acquisition. It is very difficult or even impossible to specify one value of $\rho$ or $\delta$ appropriate to the set of actual point-clouds.
- Considerable amount of calculations is required. For each data point, the tangent plane has to be determined from a set of neighbouring points, which includes the determination of the centroid of the neighbouring set, and the normal using principal component analysis.

Note that the key idea of surface reconstruction is to define a scalar function to represent the relationship between two arbitrary points in the space with the underlying surface. To overcall the aforementioned limitations, a new approach is proposed to reconstruct the surface from the given point clouds. The corresponding procedure is developed and shown in Figure 4, and the tasks involved in each step are explained as follows.

*Step 1*: It begins with the determination of a set of arbitrary points within the scope of interest. The scope of interest must include all points in the data set. Without losing the generality, this scope is defined by the boundaries along the three axes of the Cartesian coordinate system.

*Step 2*: Intensive points are generated from the vision sensors. However, it would be very inefficient to calculate the scalar distance of all the points in the defined space to the underlying surface $U$. An approach similar to the cell decomposition in a finite element method is proposed. The space of interest is decomposed into a set of finite cells based on the accuracy requirements of surface reconstruction. Each cell is represented by its central position. The vertices of a cell are called the nodes.

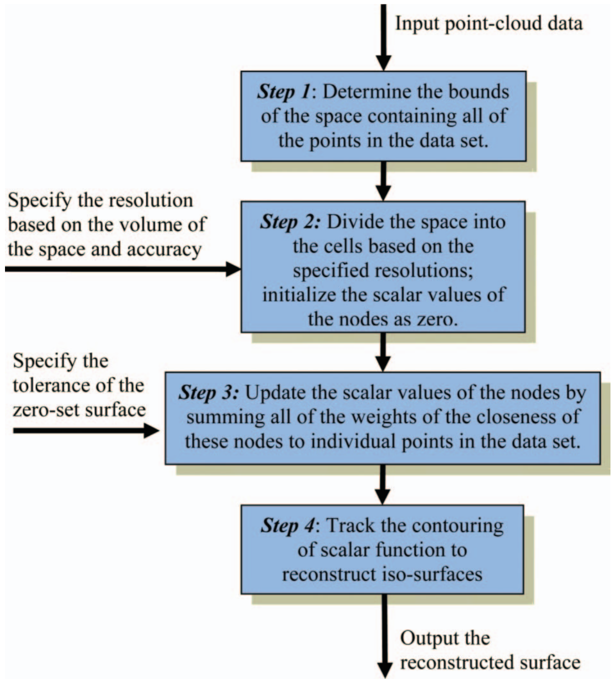



Figure 4. The procedure of the new approach for surface reconstruction.

Therefore, only the scalar distances from the nodes to the surface *U* are calculated, regardless of the number of actual data points located inside a cell.

*Step 3*: The scalar distance is defined by mapping the points of the data set to the node of each cell based on the weighted distance between each point to the concerned node. As shown in Figure 5, to minimise the calculation, the following weighted function is applied: one point has the mapped weights only on the nodes of the cell where this point is located. The weights to the nodes of any other cells are zero. The look-up tables for weights are given in Table 1. Note that the scalar value corresponding to each point is actually a measure of the probability of the node that belongs to the underlining surface; the higher value it is, the higher possibility this node is on or near the surface. From this point of view, this value is always larger or at least equal to zero. Differently from the distance function in Hoppe' algorithm, no sign is required for the scalar value based on probability.

*Step 4*: Once the scalar functions over the divided space are determined, the contouring algorithm can be applied to generate an iso-surface. The problem of extracting an iso-surface has been well-studied (Dobkin *et al.* 1990). From Steps 1 to 3, the whole scope of interest is represented by a set of nodes with corresponding scalar distances to the surface *U*. An iso-surface can be extracted by determining the zero set of the surface with the contour tracking. The contouring tracking algorithm proceeds through the scalar field, taking each cell with eight nodes at a time, and then calculating the polygon(s) for the representation of the part of the iso-surface that passes through this cell. The individual polygons are then fused into the desired surface.

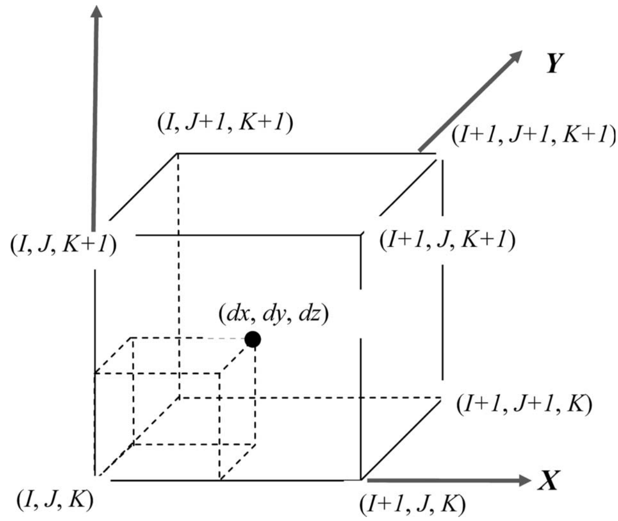



Figure 5. A data point and its weights on its neighbouring nodes.

Table 1. Look-up table of the weights.

| | Weights | | | |
|---|---|---|---|---|
| Nodes | $x$-axis | $y$-axis | $z$-axis | Total |
| $(I, J, K)$ | $(1-dx)/12$ | $(1-dy)/12$ | $(1-dz)/12$ | $(3-dx-dy-dz)/12$ |
| $(I, J, K+1)$ | | | $dz/12$ | $(2-dx-dy+dz)/12$ |
| $(I, J+1, K)$ | | $dy/12$ | $(1-dz)/12$ | $(2-dx+dy-dz)/12$ |
| $(I, J+1, K+1)$ | | | $dz/12$ | $(1-dx+dy+dz)/12$ |
| $(I+1, J, K)$ | $dx/12$ | $(1-dy)/12$ | $(1-dz)/12$ | $(2+dx-dy-dz)/12$ |
| $(I+1, J, K+1)$ | | | $dz/12$ | $(1+dx-dy+dz)/12$ |
| $(I+1, J+1, K)$ | | $dy/12$ | $(1-dz)/12$ | $(1+dx+dy-dz)/12$ |
| $(I+1, J+1, K+1)$ | | | $dz/12$ | $(dx+dy+dz/12$ |

On applying a contouring algorithm, an index to a pre-calculated array of 256 possible polygon configurations ($2^8 = 256$) within the cube is predefined. After all eight scalars on the nodes of a cell are evaluated, they can be lumped as the actual index to the polygon configuration array. If the index belongs to one of 16 cases in Figure 6, each vertex of the polygons is determined on the cube's edge by linearly interpolating the two scalar values that are connected by that edge. Note that, by using the partition of the space and the scalar values from Step 1, the computation for contouring tracking can be reduced significantly.

### *5.2.3. Validation via case studies*

To validate the effectiveness of the developed new algorithm, a comparison is made between the new algorithm and the two existing algorithms in the ITK/VTK package. In the insight segment and registration toolkit (ITK)/visualisation toolkit

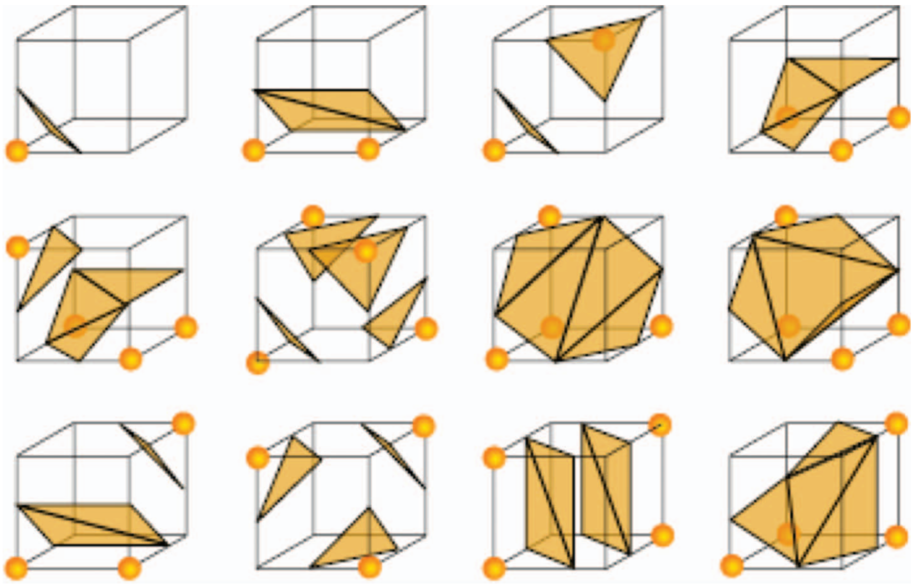


Figure 6. Sixteen special cases of contouring tracking (Lorensen and Cline 1987, Wikipedia 2011).

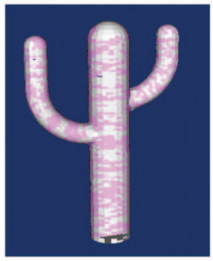
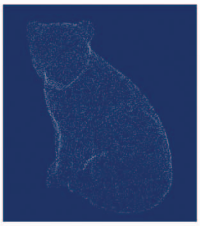
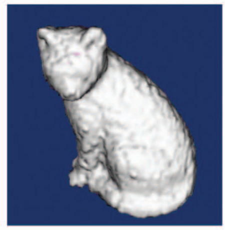
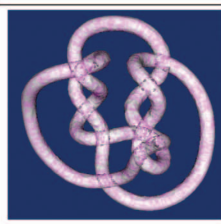
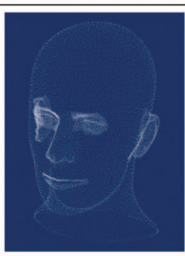
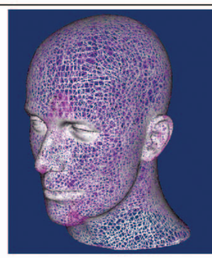
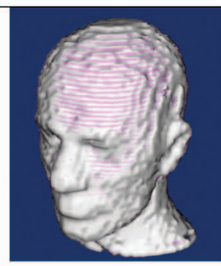
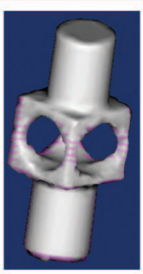
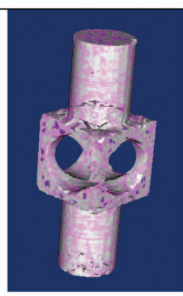
(VTK) package (Schroeder *et al.* 2004), the Hoppe's algorithm and the power crust algorithm have been implemented. Both algorithms can be applied in surface reconstruction. An overview of Hoppe's algorithm has been given in Section 5.2. The power crust algorithm for 3D surface reconstruction is based on the medial axis transform, which is guaranteed to produce a geometrically and topologically correct approximation to the surface when the data are sufficiently dense (note that in practice, the sufficient dense is very hard to be achieved). The tested data except the last one were downloaded from Hoppe's web site (Hoppe 2009). Table 2 shows the reconstructed surfaces obtained from three different algorithms.

Note that the two benchmarking algorithms were run under the default settings of the required parameters, since it is ideal for manufacturing applications to avoid manual interventions during the data process. The following conclusions can be drawn from Table 2.

(1) The new algorithm is the only one which can generate the surfaces with a correct topology and shape for all of the tested data. Both Hoppe's algorithm and power crust algorithm work well for the part with one manifold or less noise. However, when the geometric shapes become complex, the reconstructed surfaces from these algorithms are either incorrect or cannot be obtained in a reasonable computation time.

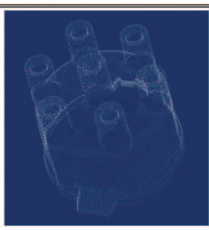
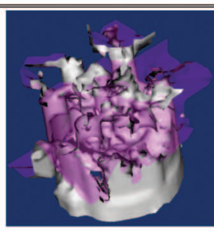
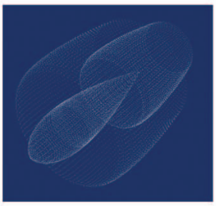
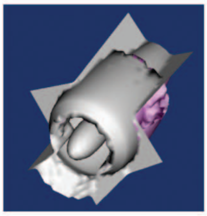
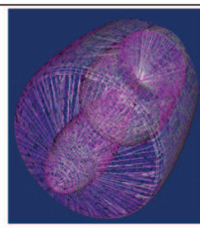
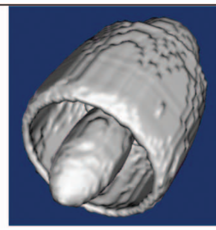
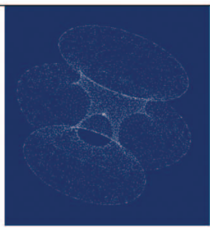
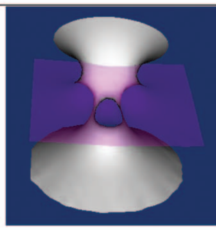
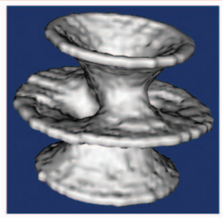
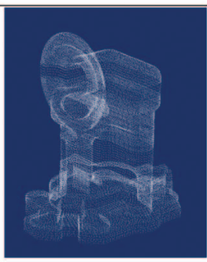
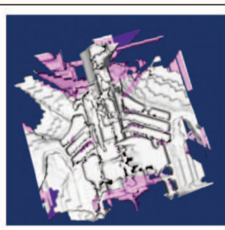
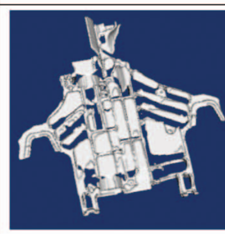
(2) By observing the processing time, the new algorithm is slightly slower when the number of points is small (e.g. 10,000 or less), but it is faster when the number of points large (e.g. 100,000 and over) in comparison with the other two algorithms. From the introduction of the new algorithm in Section 5.2.2, it is clear to see that the computation time of the new algorithm increases linearly with the number of points in the point clouds and increases

Table 2. Comparison of reconstruction results from three algorithms.

| Point – cloud data | Surface reconstruction algorithm | Power crust surface reconstruction | New algorithm |
|---|---|---|---|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

(*continued*)

Table 2. (*Continued*).

| **Point – cloud data** | **Surface reconstruction algorithm** | **Power crust surface reconstruction** | **New algorithm** |
|---|---|---|---|
|  |  |  |  |
|  |  |  |  |
|  |  | No result in a reasonable computation time |  |
|  |  | No result in a reasonable computation time |  |
|  |  | No result in a reasonable computation time |  |

exponentially with the resolution of the space partition. It is considerably advantageous compared to some other algorithms whose computation time increases exponentially with the number of points.

(3) One observed drawback of the new algorithm may be the accuracy of reconstructed surfaces. The surfaces are slightly offset with the size of a half cell and the smoothness of the reconstructed surfaces is not as good as the surfaces from the other two algorithms. These problems mainly depend on the resolution of the space partition. Note that improvement of the image quality can be made by simply increasing the resolution; however, at the expense of extended processing time. The trade-off between the accuracy and processing speed has to be made. Fortunately, for many surface applications such as painting, the machine tool usually has a considerable distance from the surface to be treated; therefore, a slightly lower accuracy or smoothness of the reconstructed surface has no impact on the task of programming the machine as long as the surface has a correct geometry and topology.

#### *5.2.4. An ITK/VTK based data processing system*

As previously discussed in Section 5.2.1, in addition to the critical task of surface reconstruction, there are many other tasks to acquire data from sensors and translate them into a CAD model for robotic programming. Numerous works have been published to address various issues in data processing from raw point-cloud data to a CAD part model or other features related to manufacturing processes. Some software tools are commercially available for very different purposes ranging from scanner controlling to 3D modelling. For example, Rocchini *et al.* (2001) have introduced a 3D scanning software suite for range maps alignment, range maps merge, mesh editing, and mesh simplification. Metron 3D scanning system provides the combination of speed, accuracy, flexibility, and ability to scan reflective surfaces that is required to meet the unfilled needs for a broad range of digital inspection applications (Johnston 2006). Besides, most producers of 3D scanners provide software tools to support data processing; however, none can be regarded as complete and satisfactory (Boehler *et al.* 2002).

To accomplish the goal for sensing and interpreting changes to reprogram reconfigurable manufacturing machines, we expand the capabilities of the ITK (Lbanez and Schroeder 2005) and the VTK (Schroeder *et al.* 2004). ITK and VTK are open-source toolkits for data processing developed by the National Library of Medicine at the National Institutes of Health. ITK focuses on data processing while VTK on visualisation. Although the primary applications of ITK/VTK are medical applications, they contain most of the well-developed algorithms for our purposes, in particular, on contouring tracking.

ITK and VTK are the libraries of algorithms and tools, rather than other ready operational systems or software environments. To access the algorithms and tools in the libraries, the user has to integrate them into the user's own application. As shown in Figure 7, a Java-based system is developed for the integration. The ITK and VTK libraries are included as the core tools to perform various tasks discussed in Section 5.2.1 and an interface is developed to read the sensor data from a physical system and export the surface models to the software of robotic programming.

As shown in Figure 8, the Eclipse platform is used (Eclipse 2003) for integration. The developed project contains all .dlls and .jars of ITK/VKT required by data processing. The project built in the eclipse platform can access C/C+ + functional libraries via java native interface. The prototyping system has been developed (Bi

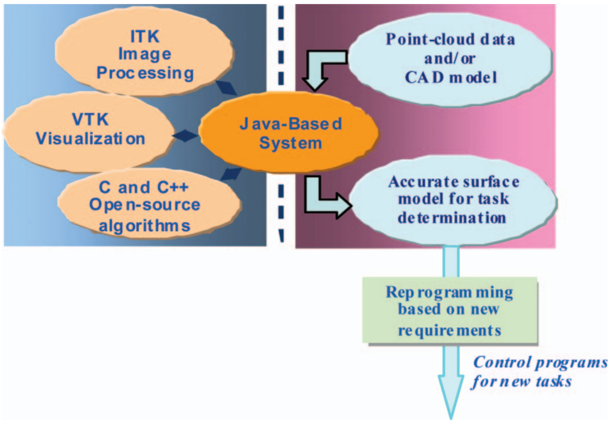



Figure 7. An ITK/VTK data processing system for sensing and interpreting changes.

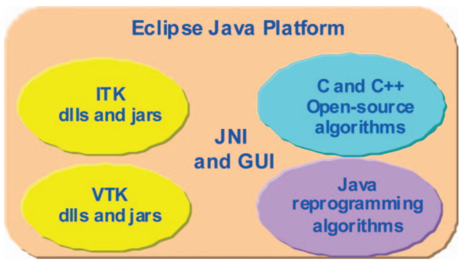



Figure 8. Implementation of the ITK/VTK based data processing system.

and Lang 2007a) and our application-related algorithms and graphic user interface (GUI) have been directly written in Java.

### *5.3. Changes in closed control loop*

To meet the task requirements, a reconfigurable machine has to be programmed and controlled to fulfil different tasks. Industrial robots have flexibility to change their control programmes and generate different motion paths. In the surface treatments such as painting, inspection, polishing, or deburring, robots have to be programmed to move around and cover the whole surfaces of the product.

An innovative approach for dealing with the changes in automation has been proposed in our previous work (Bi *et al.* 2007). The developed integrated system took feedback from the vision-sensors as inputs, created the surface model of a part to define the task requirements for machining, and generated feasible robot programmes for new tasks as outputs. As shown in Figure 9, the system consists of four basic modules, the *data-acquisition and processing* module is to generate an accurate surface model when a part is ready to accept processing. The functions of the module have been discussed in Section 5.2. The module of calculations of *working points* defines all tag points to which the robots need to cover the whole surface. The *trajectory generation* module creates robot paths so that robots can pass through all the working points for operation at the minimised cost. The *system simulation* module validates whether or not the robot paths are feasible and evaluates the efficiency and productivity of manufacturing processes. All modules have to be integrated seamlessly, except the hardware interfaces, which are required to communicate with physical devices.

In Figures 10–13, an example is provided to illustrate the procedure of robot programming for a task determined by the scanned point-cloud data. Figure 10 shows the original point-cloud data from a laser scanner. Shown in Figure 11 are the tag points on this surface model. Each tag point is represented by a tool model (painting cone). Obviously, if a robot performs the operations on all the tag points,

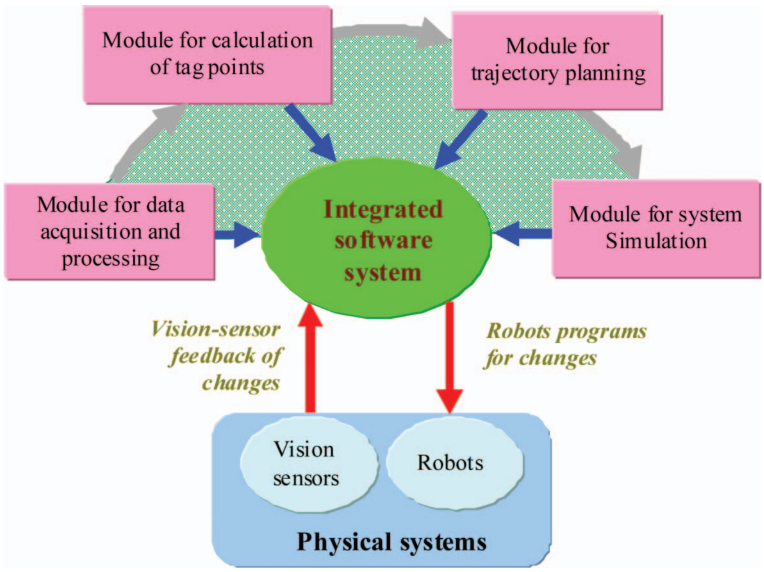



Figure 9. An integrated software system for the reconfigurable machine.

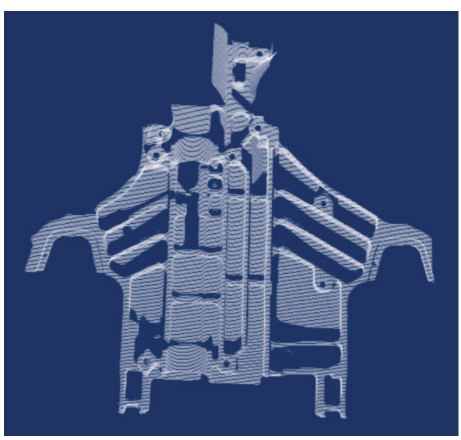


Figure 10. Point-cloud data of the part.

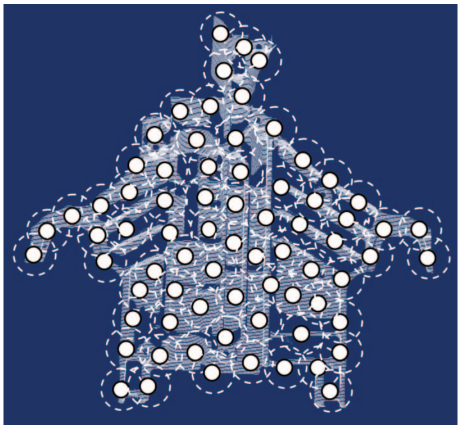


Figure 11. Generated tag points on the reconstructed surface.

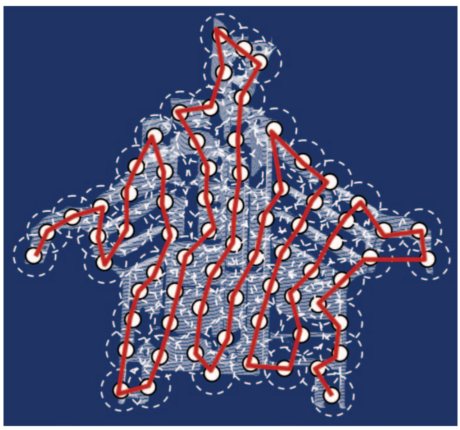


Figure 12. Generated robotic path on the surface.

Figure 13. Programming and simulation for robotic operation.

the entire surface can be treated. Figure 12 shows some robot paths represented by the red lines. Figure 13 is a snapshot of the simulation when the robot is passing through all the tag points for the painting operation on the part.

### 5.4. Conclusion

In this paper, a generic model of FMS has been proposed, which consists of the sensor-based detecting sub-systems and flexible machining systems. Technologies to acquire sensor data and interpret them for software reconfiguration have been overviewed. A new algorithm for surface reconstruction has been proposed to address the lack of reliability and inefficiency of existing algorithms. Two innovations in the proposed algorithm are (1) the space partition is applied to condense and simplify raw data from 3D laser scanners; the exponential increase in the computation time with the number of points can be alleviated and (2) the scalar distance is defined as a weighted value locally from the interested points to the nodes of the cell; a look-up table is used to accelerate the calculation. The advantages of the new algorithm are its superior capability of dealing with complexity and noise, high robustness, and a short computation time. These advantages are validated through the comparison with other existing algorithms. The disadvantage is a relatively low accuracy, which is, however, insignificant for surface applications such as painting and coating as the machine tool is usually distanced far from the surface to be treated. Note that the accuracy of surface reconstruction in the new algorithm can be improved by increasing the resolution of space partition.

As a promising application of the new algorithm for surface reconstruction, an integrated, vision sensor-based, automated coating system has been proposed. The algorithm has been applied to data acquisition and processing for surface reconstruction. An ITK/VTK based software system has been developed to generate tasks from a surface model based on the feedback from vision sensors, and an intelligent programming system has been developed to generate robot programmes for new tasks automatically. Our future work is to construct a prototypical system to validate the present theoretical studies.

## References

Abdel-Wahab, M.S., *et al.*, 2005. An enhanced algorithm for surface reconstruction from a cloud of points, GVIP 05 Conference, 119–21 December, CICC, Cairo, Egypt. Available from: www.icgst.com.

Allgower, E.L., and Schmidt, P.H., 1985. An algorithm for piecewise linear approximation of an implicitly defined manifold. *SIAM Journal of Numerical Analysis*, 22, 322–346.

Amenta, N.C., Sunghee, K., and Ravi, K., 2001. The power crust, unions of balls and the medial axis transform. *Computational Geometry: Theory and Applications*, 19, 127–153.

Azernikov, S., Miropolsky, A., and Fischer, A., 2003. Surface reconstruction of freeform objects based on multi-resolution volumetric method, *Transactions of the ASME, Journal of Computing and Information Science in Engineering*, 3, 334–337.

Bernardini, F., *et al.*, 1999. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5 (4), 349–359.

Bi, Z.M., *et al.*, 2008. Reconfigurable manufacturing systems: the state of the art. *International Journal of Production Research*, 46 (4), 967–992.

Bi, Z.M., and Lang, S.Y.T., 2007a. A framework for CAD and sensor based robotic coating automation. *IEEE Transactions on Industrial Informatics*, 3 (1), 84–91.

Bi, Z.M., and Lang, S.Y.T., 2007b. Automated robotic programming for products with changes. *International Journal of Production Research*, 45 (9), 2105–2118.
Bi, Z.M., and Wang, L., 2010. Advances in 3D data acquisition and processing for industrial applications. *Robotics and Computer Integrated Manufacturing*, 26 (5), 403–413.
Bi, Z.M., Wang, L., and Lang, S.Y.T., 2007. Current status of reconfigurable assembly systems. *International Journal of Manufacturing Research*, 2 (3), 303–327.
Bi, Z.M., and Zhang, W.J., 2001. Modularity technology in manufacturing: taxonomy and issues. *The International Journal of Advanced Manufacturing Technology*, 18 (5), 381–390.
Biegebauer, G., Pichler, A., and Vincze, M., 2002. *Detection of geometric features in ranges images for Automated Robotic Spray Painting of Lot Size One;* IROS, 2002.
Boehler, W., *et al.*, 2002. *3d scanning software: an introduction*. International Workshop on Scanning for Cultural Heritage Recording – Corfu, Greece, Sept. 1–2, 47–51.
Cazals, F., Giesen, J., and Yvinec, M., 2004. *Delaunay triangulation based surface reconstruction: a short survey*. Available from: http://hal.inria.fr/inria-00070609/en/.
Curless, B., and Levoy, M., 1996. *A volumetric method for building complex models from range images*. Proc. SIGGRAPH′96.
Dobkin, D.P., *et al.*, 1990. Contour tracing by piecewise linear approximations. *ACM Transactions on Graphics*, 9 (4), 389–423.
Eclipse, 2003. *Eclipse platform technical overview*. Available from: http://www.eclipse.org/whitepapers/eclipse-overview.pdf.
ElMaraghy, H.A., 2005. Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing Systems, Special Issue on Reconfigurable Manufacturing Systems* 17, 261–276.
Engel, K., Westermann, B., and Ertl, T., 1999. *Iso-surface extraction techniques for web-based volume visualization*. Proceedings of the conference on Visualization '99, San Francisco, CA, USA, 139–146.
Fisher, R.B., 2002. Applying knowledge to reverse engineering problems. Proceedings of Geometric Modeling and Processing, Riken, Japan, 149–155.
Freedman, D., 2004. *Surface reconstruction, one triangle at a time,* CCCG 2004, Montreal, Quebec, August 9–11, 15–19.
Gois, J.P., *et al.*, 2004. *Surface reconstruction: classification comparisons, and applications*, XXV Iberian Latin American Congress on Computational Methods – CILAMCE, 2004, Fortaleze. Proceedings CILAMCE′2004, 1–15.
Hoppe, H., 1994. *Surface reconstruction from unorganized points*. Thesis (PhD), University of Washington. Available from: http://research.microsoft.com/en-us/um/people/hoppe/thesis.pdf.
Hoppe, H., 2009. *Surface reconstruction from unorganized points*. Available from: http://research.microsoft.com/en-us/um/people/hoppe/proj/recon/.
Hoppe, H., *et al.*, 1992. Surface reconstruction from unorganized points, *ACM SIGGRAPH 1992*, 71–78.
Johnston, K., 2006. *Automotive applications of 3D laser scanning*. Available from: www.metronsys.com.
Lbanez, L., and Schroeder, W., 2005. *ITK Software guide*, ISBN 1-930934-15-7, Kitware Inc.
Li, L., 2000a. An analysis of the sources of competitiveness and performance of Chinese manufacturers. *International Journal of Operations and Production Management*, 20 (3), 299–315.
Li, L., 2000b. The effects of manufacturing capability on performance in a changing business environment: an empirical analysis of Chinese manufacturing firms. *Industrial Management and Data Systems*, 100 (5 and 6), 261–270.
Li, L., 2007. *Supply chain management: concepts, techniques and practices- enhancing value through collaboration*. Singapore: World Scientific Publishing.
Li, L., *et al.*, 2008. Assessing the effects of manufacturing infrastructure preparation prior to enterprise information systems implementation. *International Journal of Production Research*, 46 (6), 1645–1665.
Lorensen, W.E., and Cline, H.E., 1987. Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics*, 21 (4), 163–169.
Mari, F., and Mari, M., 2004. Matlab toolbox for analysis of 3D images. *Analysis*, 6, 70–77.

Mehrabi, M.G., Ulsoy, A.G., and Koren, Y., 2000. Reconfigurable manufacturing systems: key to future manufacturing. *Journal of Intelligent Manufacturing*, 11, 403–419.
Neugebauer, P.J., and Klein, K., 1997. Adaptive triangulation of objects reconstructed from multiple range images, IEEE Visualization′97 – Late-Breaking Hot Topics Session, October 1997. Available from: http://citeseer.ist.psu.edu/neugebauer97adaptive.html.
Pernkopf, F., 2005. 3D surface acquisition and reconstruction for inspection or raw steel products. *Computers in Industry*, 56, 876–885.
Rebolj, D., *et al*., 2008. Automated construction activity monitoring system. *Advanced Engineering Informatics*, 22 (4), 493–503.
Rocchini, C., *et al*., 2001. A suites of tools for the management of 3D scanned data, http://vcg.isti.cnr.it/publications/papers/3d_dima.pdf.
Schall, O., and Samozino, M., 2005. *Surface from scattered points: a brief survey of recent developments*. 1st International Workshop towards Semantic Virtual Environments, March 16–18, 2005, Villars, Switzerland, 138–147.
Schroeder, W., Martin, K., and Lorensen, B., 2004. Visualization toolkit. 3rd ed., ISBN: 1-930934-12-2. Kitware Inc.
Shih, N.J., and Wang, P.H., 2004. Point-cloud-based comparison between construction schedule and as-built progress: long-range three-dimensional laser scanner's approach. *Journal of Architectural Engineering*, 10 (3), 98–102.
Song, Y., Vergeest, J.S.M., and Bronsvoort, W.F., 2005. Fitting and manipulating freeform shapes using templates. *Journal of Computing and Information Science in Engineering*, 5 (2), 86–94.
Thompson, W.B., *et al*., 1999. Feature-based reverse engineering of mechanical parts. *IEEE Transactions on Robotics and Automation*, 15 (1), 57–66.
Várady, T., Martin, R., and Cox, J., 1997. Reverse engineering of geometric models-an introduction. *Computer Aided Design*, 29 (4), 255–269.
Vincze, M., *et al*., 2002. Automatic robotic spray painting of low volume high variant parts. Proceedings of the 33rd ISR (International Symposium on Robotics) October 7–11, 2002.
Wikipedia, Marching Cube. Available from: http://en.wikipedia.org/wiki/Marching_cubes [Accessed on 27 December 2011].
Xu, L., *et al*., 2005. A polychromatic sets approach to the conceptual design of machine tools. *International Journal of Production Research*, 43 (12), 2397–2422.
Xu, L., *et al*., 2007. A decision support system for product design in concurrent engineering. *Decision Support Systems*, 42 (4), 2029–2042.
Yin, Y., Fan, Y., and Xu, L., 2012. EMG & EPP-integrated human-machine interface between the paralyzed and rehabilitation exoskeleton. *IEEE Transactions on Information Technology in Biomedicine* 1–1. Available from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6126040 [Accessed 27 January 2012]. ISSN: 1089-7771.
Yin, Y., and Xie, J., 2011. Reconfigurable manufacturing execution system for pipe cutting. *Enterprise Information Systems*, 5 (3), 287–299.
Zhang, Y., *et al*., 2011. A new method for automatic synthesis of tolerances for complex assemblies based on polychromatic sets. *Enterprise Information Systems*, 5 (3), 337–358.
Zhao, H.K., Osher, S., and Fedkiw, R., 2001. *Fast surface reconstruction using the level set method*. The IEEE Workshop on Variation and Level Set Methods in Computer Vision (VLSM 2001), July, 2001.