Int J Adv Manuf Technol (2001) 18:313–322

The International Journal of
Advanced
Manufacturing
Technology

Determining Build Orientation for Layer-Based Machining

Z. Y. Yang, Y. H. Chen and W. S. Sze

Department of Mechanical Engineering, The University of Hong Kong, Hong Kong

Some inherent limitations exist in current layered manufacturing (LM) technologies (e.g. little choice of material, small part size, and poor surface quality) and traditional NC machining (e.g. the restriction of tool accessibility to internal features and small working area). In order to overcome these limitations, a rapid manufacturing method called robot-based layered manufacturing (RoLM) is developed. A robot with a milling cutter mounted on the end-effector is used to build a part layer by layer. Given a part model, the determination of build orientation is the first step in the manufacturing cycle and has a large effect on the surface quality and build time. In this paper, a method is proposed to determine the build orientation for RoLM by considering part accuracy and build time. Algorithms are developed to calculate tool accessibility, part stability, and the number of required support for overhangs.

Keywords: Accessibility; Build orientation; Layered Manufacturing; NC machining

1. Introduction

Part orientation has a great effect on both layered manufacturing (LM) and numerically controlled (NC) machining. Factors that influence the selection of part orientation for LM and NC machining are different. In LM, the main considerations for part build orientation include part accuracy, build time, build cost, number of supports, support-contact area, trapped volume, and part stability. In NC machining, the orientation problem can be defined as maximisation of the tool accessible surface area in one set-up. In our robot-based layered manufacturing (RoLM) approach, all of the aforementioned factors (except for trapped volume) must be considered.

1.1 Layered Manufacturing

Process planning techniques in layered manufacturing have been widely discussed [1]. A crucial step in LM process planning is to choose a preferred direction for building the model, i.e. *build orientation*. Build orientation affects the part accuracy, number of supports, support volume, and build time. It is easier to understand the importance of build orientation by studying the triangular prism in Fig. 1 where vector **d** indicates the build orientation. If the part is built as Fig. 1(*a*), the best surface finish can be obtained because no staircase effect is incurred. However, a longer time is needed to build the part because more layers are required. If it is built as Fig. 1(*b*), a staircase effect will occur on surface 1–3, but a shorter build time is required. The part can be built in the shortest time in orientation Fig. 1(*c*) because the number of layers is minimised. However, a staircase effect occurs on both surfaces 1–2 and 2–3. If the part is oriented as Fig. 1(*d*), a support structure will be required for surface 1–3 and the part is unstable, so we have to make some trade-off in the determination of build orientation. Part accuracy, build time, support structure, and part stability are the main factors for consideration. Tool accessibility is not considered in LM because the part is built incrementally by thin layers. Parts with any geometric complexity, including internal features, can be made by LM.

In current commercial systems, the build orientation is often chosen manually based on experience. In order to make this operation automatic and robust, several methods have been proposed. Majhi et al. [2,3] presented a number of optimisation algorithms to minimise staircase error, support volume, and support contact area. However, those optimisations that are related to support, can be applied only to a convex polyhedron. Puduhai and Dutta [4] optimised the build orientation by taking the number of layers and the ratio between the total staircase area and total surface area as the criteria, while the support factor is ignored. Hur and Lee [5] developed an optimisation

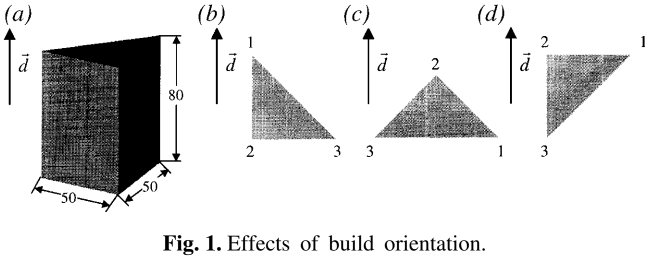

Fig. 1. Effects of build orientation.

Correspondence and offprint requests to: Dr Y. H. Chen, Department of Mechanical Engineering, The University of Hong Kong, Haking Wong Building, Pokfulam Road, Hong Kong. E-mail: yhchen@hkucc.hku.hk

method by considering part accuracy, build time, support volume, as well as part stability, but the algorithm is only applicable for stereolithography apparatus (SLA). Lan et al. [6] established some decision criteria based on surface quality, build time, and the complexity of support, to identify a desirable fabrication orientation, but the three criteria were considered independently and the algorithms were based on a constant layer thickness. Again, their approach was developed for SLA only. Some multi-objective optimisations based on CAD models instead of STL files have also been implemented. In [7], the objective functions employ weights, assigned to different surface types affecting the part accuracy. Genetic algorithms (GA) are employed in [8] to find the optimal local adaptive thickness, which is used to assess build time. The centre of gravity and the convex hull of the part are calculated to evaluate stability. In [9], build time, cost, support information, and problematic features are used as criteria to choose the best build orientation from a set of candidates. Alexander et al. [10] initiated a cost model and orientation module to evaluate the cost of a variety of orientations on different LM machines. Kulkarni et al. [1] gave a summary of the issues of build orientation in LM. Most of the strategies mentioned above are developed for a specific LM machine. None of them can be directly applied to our proposed RoLM method.

1.2 NC Machining

In NC machining, the build orientation problem is referred to as set-up determination, which is one of the main tasks in process planning. Tool-path generation is based on workpiece orientation, machine parameters, and cutter parameters [11]. Workpiece mounting and dismounting is time-consuming and leads to inaccuracy, so there should be as few set-ups as possible. That means that the cutting tool should be able to access all the surfaces to be machined in a minimum number of set-ups. Chen and Woo and colleagues [12–16] studied this topic using visibility maps, spherical geometry, and geometric dualities. Yang et al. [17] pointed out that a visibility map is not suitable for the everchanging tool-approach direction for NC tool-path planning. Instead, they used a visibility cone to analyse tool accessibility. However, their algorithm is developed for a Bezier surface only. Elber [18] simplified the accessibility problem of 5-axis machining to that of 3-axis machining; but his algorithm is based on a flat end tool, which is seldom used in freeform finishing.

Shape deposition manufacturing (SDM) is a hybrid layered manufacturing process developed jointly at Stanford University and Carnegie Mellon University [19–21]. In SDM, objects are constructed by sequential deposition and machining of material layers. However, SDM is still under development and is not available on the market yet. Although 5-axis CNC machining is considered, only 3-axis machining is employed in SDM. Many papers and dissertations on SDM have been published, but the build orientation issue has not been fully investigated.

1.3 Robot-Based Layered Manufacturing

An articulated industrial robot can carry out 5-axis machining. Tangelder et al. [22,23] established a robot-based sculpturing system and proposed a tool-path generation algorithm based on the Minkowski operation. Their system is for 3-axis machining. Ng et al. [24] developed a 5-axis (3 linear and 2 rotary) milling system for RP that can carry out tangent milling, and presented a motion compensation algorithm. However, build orientation was not considered. Horváth et al. [25] used an electronic/mechanical controlled flexible blade to manufacture a freeform front surface layer by layer to eliminate the staircase and exceeding the size limitation in commercial RP and NC machines; but the geometric complexity which can be dealt with by their system is very limited. Tse and Chen [26] developed a robotic system for machining large objects. Song and Chen [27] further proposed a feature-based tool-path generation algorithm for their robotic machining system.

In view of the inevitable staircase effect in present LM and the inherent limitation to access internal features (e.g. deep cavities) in traditional NC machining, a robot-based layered manufacturing system is proposed by Chen and Song [28]. The robotic system, as shown in Fig. 2, is composed of an ABB IRB1400 articulated robot with six-degrees-of-freedom mounted on a 2 m long linear track. With this configuration, the robot can cover a working envelope of 4 m (length) × 2 m (width) × 2 m (height). A rotary platform with clamping fixtures is installed for holding the workpiece. A ball-end milling cutter is chosen for the purpose of achieving feasible tool orientations. Because the overall process of the RoLM system has been presented in a previous paper [28], this paper will describe only the build orientation algorithm. Since the STL file is the *de facto* industrial standard in LM and with it, it is simple to perform collision detection, it is also chosen as the input of the robotic machining system. In the later parts of the paper, an STL file is treated as a polyhedron.

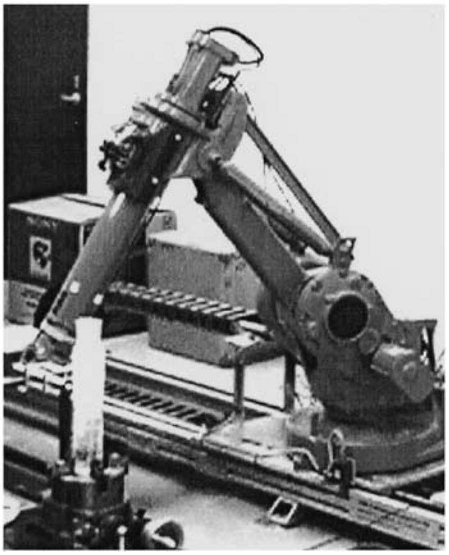

Fig. 2. Robot machining system.

2. Criteria for Build Orientation

2.1 Inaccessible Facets

In RoLM, the definition of accessibility is quite different from that defined in NC machining. The model in Fig. 3 is a good example. If it is machined with an NC machine, the hollow part will be specified as an inaccessible feature. In RoLM, the inaccessibility is resolved as in Fig. 3(*c*) by slicing the model into layers and then machining it layer by layer. In order to reduce the complexity of collision detection, we limit the tool feasible orientation (TFO) in one set-up within a subsphere as in Fig. 4(*a*), which is a subset of the working area of the robot. Once the robot and rotary platform are configured, TFO is defined. The Gaussian map (G-Map) of TFO is shown in Fig. 4(*b*), where $\mathbf{N}_b$ is the build orientation, $\mathbf{N}_t$ indicates tool feasible orientations. In Fig. 4(*c*), the robot will have difficulty in achieving such postures, and may gouge the machined surface. So the facets that satisfy the following inequality will be defined as inaccessible facets:

$$\mathbf{N}_b \cdot \mathbf{N}_f > \cos\alpha \tag{1}$$

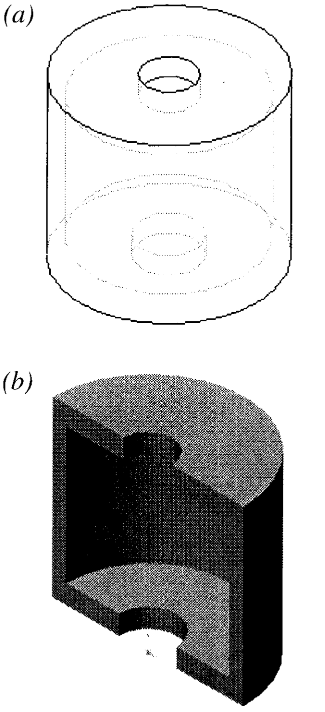

Fig. 3. Increased accessibility in RoLM. (*a*) Frame view. (*b*) Section view. (*c*) One layer in RoLM.

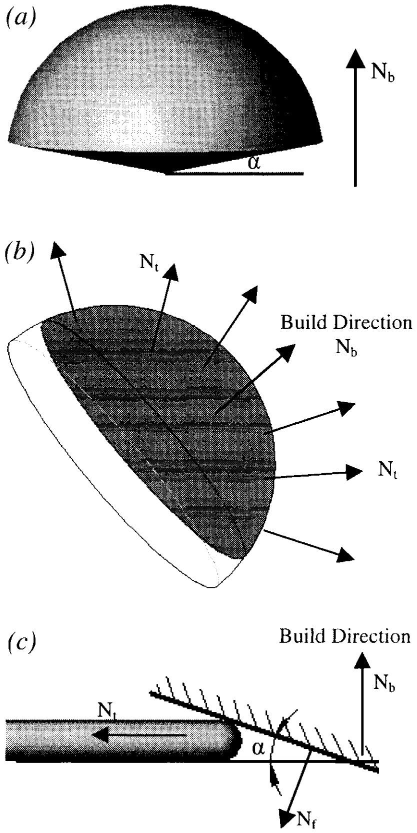

Fig. 4. G-Map of tool feasibility in one set-up. (*a*) Tool feasibility sub-sphere. (*b*) Tool feasibility G-Map. (*c*) Inaccessible facet.

where $\mathbf{N}_b$ is the normalised vector indicating the build orientation, $\mathbf{N}_f$ is the facet normal, and α is a small angle preset by the users. Inaccuracy occurs when an inaccessible facet is being machined. So, we should choose the build orientation that has the minimum F_{AE},

$$F_{AE} = 1 - \sum_i A_i/A_{Total} \tag{2}$$

where A_i is the area of the ith inaccessible facet, and A_{Total} is the surface area of the model. F_{AE} gives the percentage of the area of accessible facets over the model surface area, and $F_{AE} \in [0,1]$. The calculation of F_{AE} can be carried out in a time of $O(n)$, where n is the number of facets in the model.

2.2 Accessibility of a Point

The tool cannot approach certain points on the model surface in all the feasible tool orientations because of the existence of local interference and global interference, as shown in Fig. 5. An accessible tool orientation for a point is the tool feasible orientation in which the point can be machined without any interference between part and tool assembly. The tool can access the point **P** only from the "outside" of the tangent plane of the surface at point **P**, as illustrated in Fig. 5(*a*). This defines the point local accessible orientation (PLAO), whose G-map is a hemisphere. The vector from the centre of the

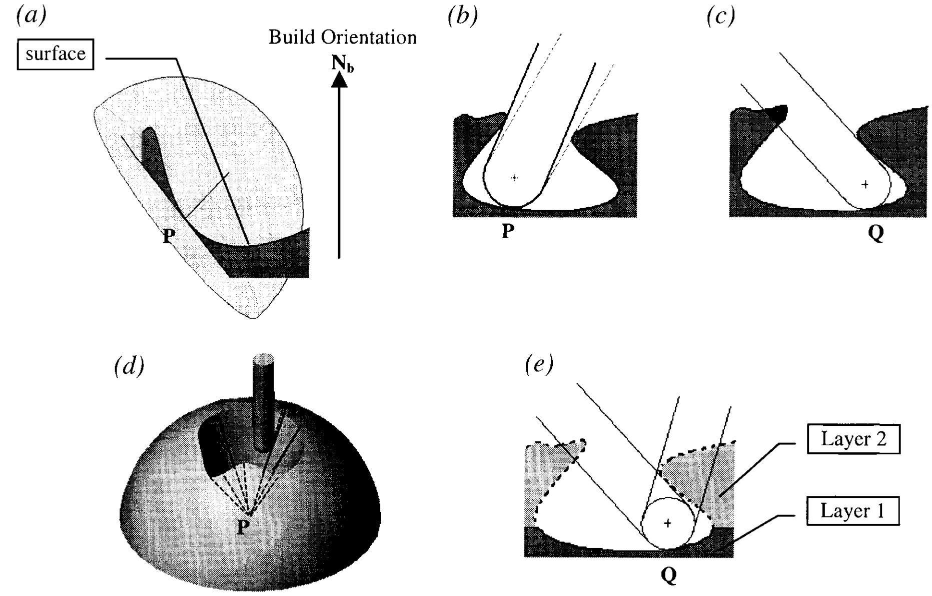

Fig. 5. Local interference and global interference of a point. (*a*) Point local accessible hemisphere. (*b*) Accessibility limited by global interference. (*c*) Inaccessible point. (*d*) Accessible cone of a point. (*e*) PGAO increased in layered machining.

hemisphere to the pole is the facet normal. Figure 5(*b*) shows that the PLAO of point **P** is limited by the interference between the cutter and the other facets. Such interference is called global interference. Figure 5(*c*) shows that point **Q** cannot be machined because global interference cannot be avoided, even by adjusting the tool orientation. Global interference free orientation of a point is called point global accessible orientation (PGAO). The intersection of TFO, PLAO, and PGAO gives all the accessible tool orientations for a given point and gives the system configuration. This forms an open cone, as shown in Fig. 5(*d*). The accessibility of a point is defined as follows.

Definition 1. The accessible subsphere of a point is the intersection of vectors of TFO, PLAO, PGAO and a unit sphere. The accessibility of point **P** (AP) is the area of its accessibility subsphere measured on the sphere.

The larger the AP is, the more likely it is that point **P** can be machined without any interference. A different build orientation defines a different AP. Therefore for orientation determination, the sum of *AP* must be maximised, that is,

$$Sum_{AP} = \sum_{i} AP_i \tag{3}$$

Sum_{AP} is then normalised using the following equation:

$$F_{AP} = Sum_{AP}/Num_{SP} \tag{4}$$

where Num_{SP} is the number of total sample points. F_{AP} reflects the number of accessible points, and F_{AP} [0,1]. The algorithm to calculate *AP*, Sum_{AP}, and Num_{SP} will be given in Section 4.

We also note that TFO is defined by the system configuration while PLAO is defined by the model geometry. They remain unchanged when the part is reoriented; but a different build orientation defines a different PGAO for each point. PGAO varies with different layer thickness in layered machining and does not decrease as layer thickness decreases. That means we can enlarge AP by layered machining. As shown in Fig. 5(*e*), point **Q** can be machined easily by considering only layer 1. That is the theoretical basis of layered machining.

2.3 Support Number

Apart from thermal deformation, the part accuracy is compromised mainly by the staircase effect in traditional LM. The reason is that zero-order approximation is used for all those facets that are neither parallel to, nor vertical to, the build orientation. In RoLM, the surface quality is decided by the scallop height between tool paths rather than the stock layer thickness. The inaccessible facet is the primary source of inaccuracy. The secondary error source is the support structure because the model surface touching the support will be scratched when the support is removed. The support structure also has the possibility of reducing point accessibility. Build time will increase as the number of supports increases because a larger area must be machined. Therefore, the total number of supports (Sum_{NS}) should be minimised when we choose the preferred build orientation. Because a special support structure is designed for RoLM, we can use the Sum_{NS} to estimate the support contact area. This will be discussed in detail in Section 5. Sum_{NS} is then coded as F_{NS} using the following equation:

$$F_{NS} = 1 - \frac{Sum_{NS}}{\sum Sum_{NS}} \tag{5}$$

where $F_{NS} \in$ [0,1].

2.4 Build Orientation Determination Strategy

We use the following function to evaluate a given orientation, and choose the best one from the candidates:

$$BO = w_e F_{AE} + w_a F_{AP} + w_s F_{NS} \tag{6}$$

where w_e, w_a, $\mathrm{w}_s \in [0,1]$ are the weights assigned by the user to the three factors, respectively. $w_e + w_a + w_s = 1$. $BO \in [0,1]$.

The candidate orientations are sorted by corresponding *BO*, and the one with largest *BO* is preferred.

3. Candidate Build Orientation Selection

We must choose some build orientations as candidates because the whole search space is continuous and unlimited. The following orientations are possible candidates and their *BO* are calculated:

1. *User defined orientations*: This can be the orientation chosen intuitively by an experienced operator.
2. *Positive Z-axis used in modelling the part*: This reflects the designer's logic.
3. *The largest face of convex hull (CH) as the bottom plane.*
4. *Convex Hull face with smallest HCG as the bottom plane*: HCG is the height of Centre of Gravity (CG) of the model.
5. *The orientation with minimum RHC*: RHC is the ratio between HCG and the corresponding CH face. RHC can be found in time $O(n)$ when CG is found.

Unless otherwise specified by the users, the last three kinds of orientation are chosen as candidate orientations by the system because they can provide better stability.

CH can be found in time $O(n\log n)$. In order to calculate the CG, the model is sliced into voxels by a 3D regular grid whose directions are parallel to the coordinate axes, respectively. The coordinates of the centre of the *i*th voxel are calculated and stored as (x_i, y_i, z_i), and its CG $(\bar{x}, \bar{y}, \bar{z})$ can be calculated as:

$$\bar{x} = \frac{1}{V}\sum_{i=1}^{V} x_i, \quad \bar{y} = \frac{1}{V}\sum_{i=1}^{V} y_i, \quad \bar{z} = \frac{1}{V}\sum_{i=1}^{V} z_i \tag{7}$$

where V is the number of voxels. The calculation is time-consuming, but the calculation time can be reduced to $O((B/r)^2 n)$ by noticing the continuity within the model, where $B = \mathrm{Max}\{X, Y, Z\}$ (X, Y, and Z are the length, width, and height of the coordinate-axis-parallel minimum bounding box of the model, respectively), r is the grid resolution, and n is the number of facets. The basic calculation is to find the intersection point of a line and a plane. The detail of the algorithm is omitted here. A smaller r gives a more accurate result, but takes longer.

4. Tool Accessibility

According to Definition 1, AP is the area of a subsphere. It is neither efficient nor necessary to aquire the accurate value. In preparation for the computation of AP, we discretise the unit sphere by refining the triangulation of its inscribed 20-side polygon (which is the polygon with the maximum number of sides). Assume that the refined triangulation (RT) has v vertex and is stored in an array named $Array_{vertex}$. Then we define $\mathrm{TFO_d}$, $\mathrm{PLAO_d}$, $\mathrm{PGAO_d}$, and $\mathrm{AP_d}$ as integers indicating the number of vertices within their corresponding subsphere on the RT. For example, the AP shown in Fig. 6 can be counted, it is 23.

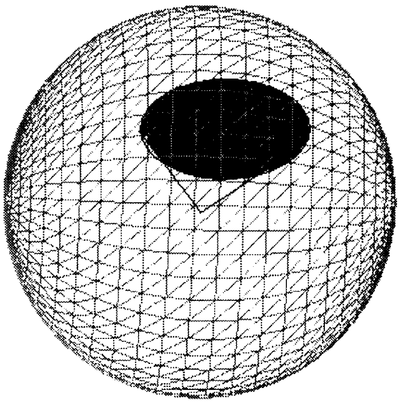

Fig. 6. Discrete representation of AP.

After the discrete representation is given, we can compute AP using the following equation:

$$\mathrm{AP_d} = (\mathrm{TFO} \cap \mathrm{PLAO} \cap \mathrm{PGAO})_\mathrm{d} \tag{8}$$

As shown in Fig. 7(*a*), TFO is a spherical cap and PLAO a hemisphere,

$$\begin{aligned}&\text{Area of TFO} \cap \text{PLAO} = \\ &\text{Area of TFOC} \cap \text{PLAO} - \text{Area of Belt} \cap \text{TFOC} \cap \text{PLAO}\end{aligned} \tag{9}$$

where TFOC is the G-Map of the TFO when $\alpha = 0$. Belt is the subsphere of TFOC − TFO. TFOC∩PLAO is a spherical biangule, and its area is:

$$A_{tp} = 2R^2\theta = 2\theta \tag{10}$$

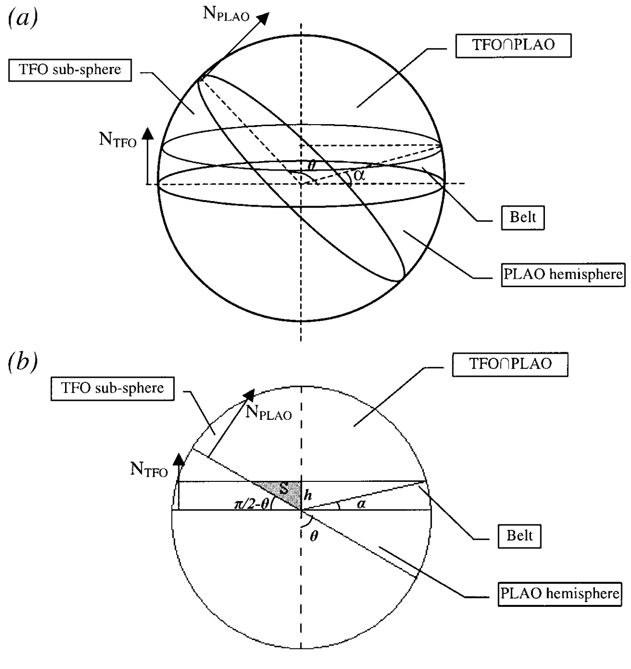

Fig. 7. Computation of TFO∩PLAO.

where R is the spherical radius (here, it is 1), θ is the angle in radians between the normal vectors corresponding to TFOC and PLAO. θ is given by:

$$\theta = \mathrm{arc\ cos}\ (\mathbf{N}_{TFO} \cdot \mathbf{N}_{GLAO}) \tag{11}$$

where $\mathbf{N}_{TFO}$ and $\mathbf{N}_{GLAO}$ are the normalised vectors corresponding to TFOC and PGAO, respectively.

However, the Area of Belt∩TFOC∩PLAO is difficult to compute because it is a triangular patch on the sphere. Since α is relatively small (no more than 5°), we can approximate it by projecting the belt onto plane T, defined by the cross-product of $\mathbf{N}_{TFO}$ and $\mathbf{N}_{GLAO}$ as illustrated in Fig. 7(b). Then we obtain:

$$\text{Area of Belt}\cap\text{TFOC}\cap\text{PLAO} = (\text{Area of Belt})/2 + 2A_s \tag{12}$$

where A_s is the area of triangle S,

$$A_s = (R\ \sin\alpha)(R\ \sin\alpha\tan\theta)/2 = \sin^2\alpha\tan\theta \tag{13}$$

$$\text{Area of Belt} = 2\pi R\ h = 2\pi\ R\ R\ \sin\alpha = 2\pi\sin\alpha \tag{14}$$

Substituting Eqs. (8), (10), (11), and (12) into Eq. 7, we obtain:

$$A_{TP} = 2\theta - (\sin^2\alpha\ \tan\theta/2 + 2\pi\ \sin\alpha) \tag{15}$$

and the discrete representation $(\text{TFO}\cap\text{PLAO})_\text{d}$ is:

$$(\text{TFO}\cap\text{PLAO})_\text{d} = \text{int}\left(\frac{A_{TP}}{4\pi}\,\nu\right) = \text{int}\left(\frac{2\theta - ((\sin^2\alpha\ \tan\theta)/2 + 2\ \sin\alpha)}{4\pi}\right) \tag{16}$$

where ν is the number of vertices on the discrete unit sphere, int() is the function giving the floor integer of a real number.

It is time-consuming to calculate PGAO in traditional NC machining; but it can be simplified in RoLM dramatically, as shown in Fig. 8. We calculate AP without computing PGAO explicitly as in the following steps:

1. Find the diameter D and the centre (midinterval point of diameter) of the model. Draw a sphere at the centre with a diameter $D + C$ to enclose the whole model. C is a large enough real number.
2. Map the remained vertex in $Array_{vertex}$ filtered by TFO and PLAO onto the new sphere (P, Q, and A_i, $i = 1, 2, \ldots, 9$ in Fig. 8).

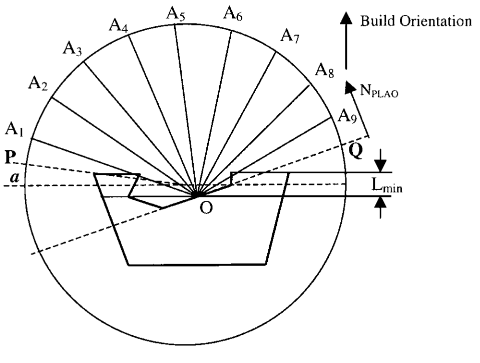

Fig. 8. Computation of PGAO_d.

3. Slice the model with plane $Z = Z_o + L_{min}$ (Z_o is the Z-coordinate of point O, L_{min} is the minimum stock layer thickness), to obtain the contour interface.
4. Find the intersection points $P_{intersec}$ between segment A_iO and the contour interface. Record the number of $P_{intersec}$ as $N_{intersec}$.
5. AP of point O is:

$$\text{AP} = (\text{TFO}\cap\text{PLAO})_\text{d} - N_{intersec} = \text{int}\left(\frac{A_{TP}}{4\pi}\,\nu\right) - N_{intersec} \tag{17}$$

Before calculating Sum_{AP}, the STL file is retriangulated to sample the model surface. Large triangles will be divided into smaller ones with a similar area and shape. The centroid of each facet is used as a sample point. The number of sample points is the aforementioned Num_{SP}. This can be carried out in time $O(n)$. Then we obtain:

$$Sum_{AP} = \sum_i |AP_i| \tag{18}$$

$$\text{where } |AP_i| = \begin{cases} 0 & AP_i \le t \\ 1 & AP_i > t \end{cases} \tag{19}$$

t is an integer ($t \ge 1$) defined by users. AP_i is normalised so as to let Sum_{AP} reflect how many POINTs can be machined, rather than how many accessible tool orientations there are for all the sample points. The build orientation with a larger Sum_{AP} indicates that more points can be machined.

5. Support Number

Because of the stiffness of the raw material, not all the overhangs need support. As illustrated in Fig. 9(a), overhangs in RoLM can be classified into two kinds according to their geometric properties: *Complete-Overhang* and *Semi-Overhang* (including cantilevered overhang). The former is *support-needed overhang*, whereas the latter can be divided further into *support-needed overhang* and *support-needless overhang*. Figures 9(b) and 9(c) give some possible support structures. Support structure for a semi-overhang used in RoLM is aimed at preventing the part from deforming owing to gravity and cutting force.

5.1 Finding Overhang

Before searching for an overhang region, the STL file is preprocessed by adding 3 pointers to each facet that indicate its 3 neighbour facets (the 3 facets sharing an edge). This can be carried out in time $O(n^2)$, where n is the number of facets. Let $f_\text{neighbour}$ denote the neighbouring facet to facet f. We call set $F_\text{neighbour}(f) = \{f^i\}$ the *neighbourhood* of facet f, if:

$$\begin{aligned} &\forall f^0 \in F_\text{neighbour}(f), \quad \exists f^0_\text{neighbour} \in \text{F}_\text{neighbour}(f) \\ &F_\text{neighbour}(f_i) = F_\text{neighbour}(f_j), \quad \text{where } f_i, f_j \in F_\text{neighbour}(f) \end{aligned} \tag{20}$$

Edge e is called a boundary edge (BE) if it is not shared by two facets in $F_\text{neighbour}(f)$. The facet with BE is called a boundary facet (BF). Facet f_p is called a perifacet (PF) if it

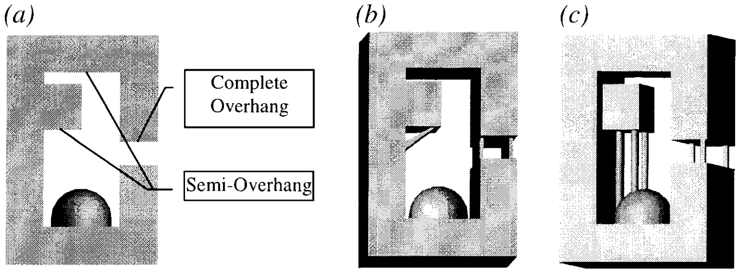

Fig. 9. Support structures.

has a BE but $f_p \notin F_{\text{neighbour}}(f)$. The vertex of PF not on the boundary edge is called a free vertex (FV). Facet $f \in F_{\text{neighbour}}(f)$ is called an inner facet (IF) if none of its edges is BE. The aggregates of BE, BF, PF, FV, and IF are also called boundary edge, boundary facet, perifacet, and free vertex of the facet neighbourhood, respectively. Figure 10(*a*) shows the BE, BF, PF, FV, and IF of the facet neighbourhood enclosed by the dashed line.

Definition 2. $F_{\text{neighbour}}(f)$ is a simple overhang (SOH) if:

1. $\forall f^o \in F_{\text{neighbour}}(f)$, $\mathbf{N}_b \cdot \mathbf{F}_f{}^o < 0$
2. Any line perpendicular to the build orientation intersects $F_{\text{neighbour}}(f)$ no more than twice, or an infinite number times if $\mathbf{N}_b \cdot \mathbf{N}_f{}^o = -1$.

BE, BF, and PF of $F_{\text{neighbour}}(f)$ are called boundary edge, boundary facet, and perifacet of SOH, respectively.

Let S_1 and S_2 denote the two half-planes defined by a given facet f, one of its neighbour facets f_n and their common edge *ce*. We call the march from f to f_n a horizontal march (H-march) if any line in S_2 which is perpendicular to *ce* is perpendicular to the Z-axis. The angle between S_1 and S_2, δ, is called the critical angle of facet f referred to edge *ce*. If $\mathbf{N}_f$ and $\mathbf{N}_{ce}$ represent the normal vectors of f and the normalised vector is defined by *ce*, $\mathbf{N}_{f_n}$ can be calculated by:

$$\mathbf{N}_{f_n} = \mathbf{Z} \times \mathbf{N}_{ce} \times \mathbf{N}_{ce} \tag{21}$$

The critical angle can be calculated by:

$$\delta = \cos^{-1}\left(\frac{\mathbf{N}_f \cdot \mathbf{N}_{f_n}}{|\mathbf{N}_{f_n}|}\right) \tag{22}$$

When we are marching from facet f to its neighbouring facet f_v, if the angle between them has $\omega < \delta$, we call the march a V-march. In Fig. 10(*b*) the march from f to f_h is an H-march, while the march from f to f_v is a V-march.

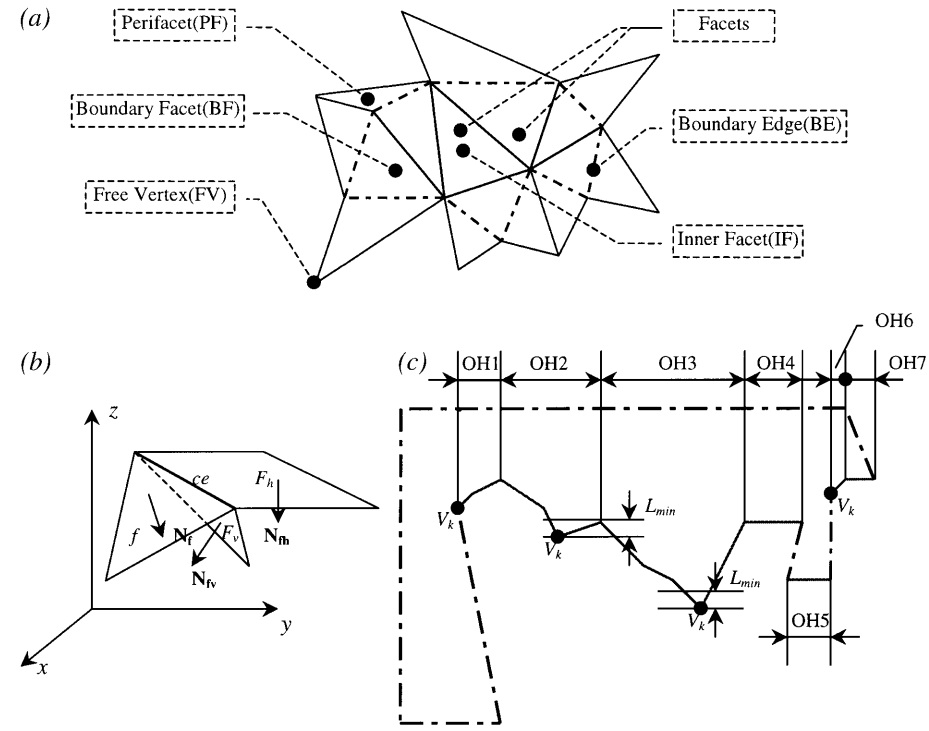

Fig. 10. Determination of overhang. (*a*) Facet neighbourhood, boundary edge, boundary facet, perifacet, and free vertex. (*b*) H-march and V-march. (*c*) Classification of overhangs.

According to its definition, overhang can be found by the following steps:

1. Orient the STL model in the candidate orientation under consideration, i.e. let the *Z*-axis in the model coordinate system be parallel to the candidate build orientation. This can be carried out in time $O(n)$.
2. Find all the facets satisfying $\mathbf{N}_b \cdot \mathbf{N}_f < 0$, where $\mathbf{N}_b$ is the normalised vector which indicates the build orientation, and $\mathbf{N}_f$ is the facets normal. Store these "downward" facets in a list F, note that each facet has the information about its neighbouring facets. This can be carried out in time $O(n)$. Then remove the facets on the bottom plane from F.
3. Find all the facets in F perpendicular to the build orientation. Group them into a facet neighbourhood and store them as $\mathrm{SOH}_\mathrm{j}^\mathrm{Horizontal}$ ($j = 1,2,...,J$). Remove them from F. This can be carried out in time $O(n)$.
4. Find a facet f with the lowest vertex V_k (vertex with minimum Z_{Vk}, which is the z-coordinate) of the facets in F. Construct $F^\mathrm{i}_\mathrm{neighbour}(f)$ by marching the facets starting from f and ending with V-march or until no neighbouring facet can be found. $F^\mathrm{i}_\mathrm{neighbour}(f)$ is stored as SOH_i. Remove all the facets in $F^\mathrm{i}_\mathrm{neighbour}(f)$.
5. If F is not empty, i++, repeat step 4.

Now all the facet neighbourhoods corresponding to overhangs are found and divided into simple overhangs: SOH_i and $\mathrm{SOH}_j^\mathrm{Horizontal}$. Figure 10(*c*) gives a 2D illustration of the overhang classification. The 7 overhangs found are shown by solid line segments. Among them OH4, OH5, and OH7 belong to $\mathrm{SOH}_j^\mathrm{Horizontal}$, while the other four belong to SOH_i. Then, we must classify them further into complete-overhang and semi-overhang by the following method.

```
For $\mathrm{SOH}_j^\mathrm{Horizontal}$, do
for($j = 1$; $j <=$ J; $j$++){
  if (FVs of $\mathrm{SOH}_j^\mathrm{Horizontal}$ are above the plane containing $\mathrm{SOH}_j^\mathrm{Horizontal}$)
    $\mathrm{SOH}_j^\mathrm{Horizontal}$ is complete-overhang (e.g. OH5);
      (Overhang Type 1, OT1 in brief)
  else{
    $\mathrm{SOH}_j^\mathrm{Horizontal}$ is semi-overhang (e.g. OH4, OH7);
    for(all the BEs of $\mathrm{SOH}_j^\mathrm{Horizontal}$){
      if(FV of BE is below the plane containing $\mathrm{SOH}_j^\mathrm{Horizontal}$){
        BE is marked as Lower BE (LBE);
      else
        BE is marked as Upper BE (UBE);
    }
  }
}
```

While for SOH_i, let Z^B_{min} denote the *Z*-coordinate of the lowest vertex on the boundary of SOH_i, and Z^V_{min} denote the *Z*-coordinate of the lowest vertex in SOH_i. SOH_i is a complete-overhang (overhang type 2, OT2) if:

$$Z^B_{min} - Z^V_{min} \geq L_{min} \tag{23}$$

where L_{min} is the minimum stock layer thickness (e.g. OH3), otherwise it is a semi-overhang (e.g. OH1, OH2, and OH6).

Now all the overhangs including SOH_i and $\mathrm{SOH}_j^\mathrm{Horizontal}$ are classified into complete-overhangs and semi-overhangs. All the complete-overhangs are support-needed, whereas semi-overhangs require support only if:

$$\begin{cases} LLBE/L < T_l & \text{for } SOH_j^{Horizontal} \quad (e.g.\ OH_4) \quad (24a) \\ ASH/PASH < T_a & \text{for } \quad SOH_i \quad (\text{e.g. } OH_6) \end{cases}$$

$$(24b)$$

where *LLBE* is the length of the lower boundary edges of the semi-overhang, *L* is the length of the overall boundary edge. *PASH* is the projection area of the semi-overhang onto the bottom plane; *ASH* is the area of the semi-overhang. A support-needed semi-overhang satisfying (24*a*) is called overhang type 3 (OT3), and that satisfying (24*b*) is called overhang type 4 (OT4). T_l and T_a are specified by the user. They are determinated according to the stiffness of the material, the cutting force, and the layer thickness.

Now, all the support-needed overhangs are classified into 4 types, OT1 and OT2 are complete-overhangs, and OT3 and OT4 are support-needed semi-overhangs. The number of supports required for OT1 is proportional to the area of the overhang. The number of supports for overhang N_s is estimated as follows:

$$N_s = \begin{cases} A & \text{for} \quad OT1 \\ c_2 PA & \text{for} \quad OT2 \\ c_3 A & \text{for} \quad OT3 \\ c_4 PA & \text{for} \quad OT4 \end{cases} \tag{25}$$

where A is the area of the overhang, PA is the projection area of the overhang onto the bottom plane. c_2, c_3, and c_4 are equivalent coefficients assigned by users. c_2, c_3, and $c_4 \in (0, 1)$. Then we can calculate the Sum_{NS}:

$$Sum_{NS} = \sum_i N_{si} \tag{26}$$

$Sum_{NS} < A_{Total}$, A_{Total} is the surface area of the model.

6. Examples

Two examples are used to demonstrate the performance of the proposed strategy. Figures 11(*a*) and 11(*b*) show the two models that are frequently used in similar research [6,29]. The candidate build orientations chosen for the models are shown in Fig. 12. Calculation results are given in Table 1, where 0.3333 is assigned to all the weightings w_e, w_a, w_s, which means that F_{AE}, F_{AP} and F_{NS} are treated equally. 5° is assigned to α. T_l is 0.5, and T_a 0.26 ≈ cos(5π/12) t in Eq. (19) is 8. $c_2 = 0.1$, $c_3 = (1 - LLBE/L)$, and c_4 0.5.

According to Table 1, none of the candidate orientations produces inaccessible facets in RoLM. Candidate orientations (*a*) and (*c*) are selected for the bracket and crank-slider base, respectively.

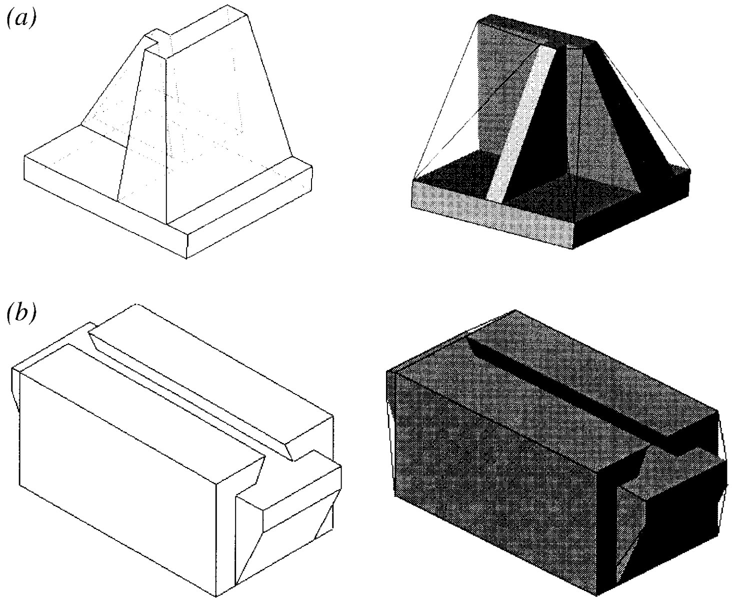

Fig. 11. Example models and convex hulls. (*a*) Bracket. (*b*) Crank-slider base.

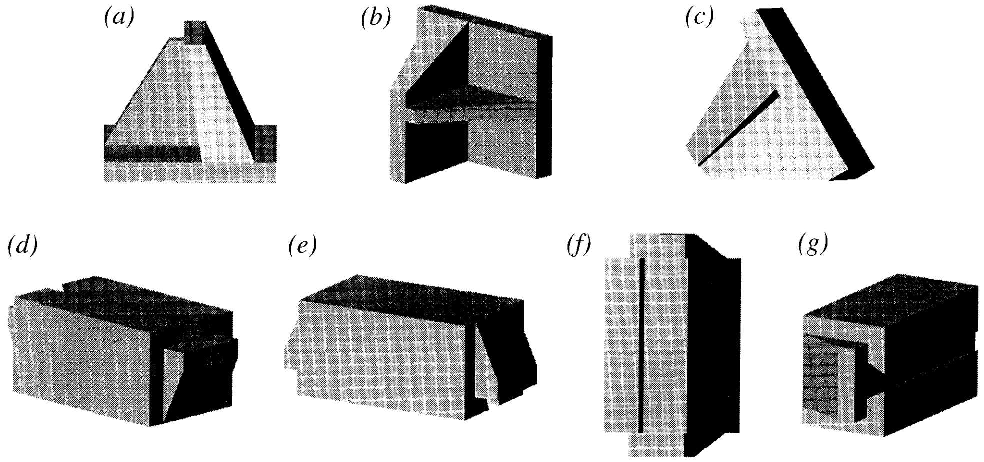

Fig. 12. Candidate build orientations. (*a*) and (*e*) Best.

Table 1. Example calculations.

Build orientation	Bracket			Crank-slider base			
Criterion	*a*	*b*	*c*	*d*	*e*	*f*	*g*
Sum_{AP}	4583.0	4583.0	3553.0	3479.0	3652.0	3514.0	3566.0
Sum_{NS}	0	3015.9	4425.6	0	0	1662.0	0
F_{AP}	1	1	0.7752	0.9526	1	0.9622	0.9765
F_{NS}	1	0.5947	0.4053	1	1	0	1
F_{AE}	1	1	1	1	1	1	1
BO	1	0.8649	0.7268	0.9842	1	0.6541	0.9922
Rank	1 (Best)	2	3	3	1(Best)	4	2

7. Conclusion

In this paper, we have presented an algorithm for evaluating build orientation. The main consideration is focused on part accuracy, tool accessibility, and number of supports.

Build orientation for RoLM is a new problem in which improvements can be made. In future research, build time will be added as a criterion. The support structure must be designed and constructed automatically. Coefficients, such as T_l and T_a, must be calculated from the physical properties of the material and system configuration.

Acknowledgement

This project is supported by a grant from the Hong Kong Grant Council, account number HKU567/96E.

References

1. P. Kulkarni, A. Marsan and D. Dutta, "A review of process planning techniques in layered manufacturing", Rapid Prototyping Journal, 6(1), pp. 18–35, 2000.
2. J. Majhi, R. Janardan, M. Smid and J. Schwerd, "Multi-criteria geometric optimization problems in layered manufacturing", SCG 98, Minneapolis Minnesota, USA, 1998.
3. J. Majhi, R. Janardan, M. Smid and P. Gupta, "On some geometric optimization problems in layered manufacturing", Computer Geometry: Theory and Application, 12, pp. 219–239, 1999.
4. N. S. Puduhai and D. Dutta, "Determination of optimal orientation based on variable layer thickness in layered manufacturing", Technical Report UM-MEAM-94–14, Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI.
5. J. Hur and K. Lee, "The development of a CAD environment to determine the preferred build-up direction for layered manufacturing", International Journal of Advanced Manufacturing Technology, 14, pp. 247–254, 1998.
6. P. T. Lan, S. Y. Chou, L. L. Chen and D. Gemmill, "Determining fabrication orientation for rapid prototyping with stereolithography apparatus", Computer-Aided Design, 29(1), pp. 53–62, 1997.
7. W. Cheng, J. Y. H. Fuh, A. Y. C. Nee, Y. S. Wong, H. T. Loh and T. Miyazawa, "Multi-objective optimization of part-building orientation in stereolithography", Rapid Prototyping Journal, 1(4), pp. 12–23, 1995.
8. F. Xu, Y. S. Wong, H. T. Loh, J. Y. H. Fuh and T. Miyazawa, "Optimal orientation with variable slicing in stereolithography", Rapid Prototyping Journal, 3(3), pp. 76–88, 1997.
9. D. T. Pham, S. S. Dimov and R. S. Gault, "Part orientation in stereolithography", International Journal of Advanced Manufacturing Technology, 15, pp. 674–682, 1999.
10. P. Alexander, S. Allen and D. Dutta, "Part orientation and build cost determination in layered manufacturing", Computer-Aided Design, 30(5), pp. 343–356, 1998.
11. K. Marciniak, Geometric Modelling for Numerically Controlled Machining, Oxford University Press, 1991.
12. L. L. Chen and T. C. Woo, "Computational geometry on the sphere with application to automated machining", Journal of Mechanical Design, 114, pp. 288–295, 1992.
13. K. Tang, T. C. Woo and J. Gan, "Maximum intersection of spherical polygons and workpiece orientation for 4- and 5-axis machining", Journal of Mechanical Design, 114, pp. 477–485, 1992.
14. L. L. Chen, S. Y. Chou and T. C. Woo, "Separating and intersecting spherical polygons: computing machinability on three-, four-, and five-axis numerically controlled machines", ACM Transactions on Graphics, 12(4), pp. 305–326, 1993.
15. K. Tang, L. L. Chen and S. Y. Chou, "Optimal workpiece setups for 4-axis numerical control machining based on machinability", 37, pp. 27–41, 1998.
16. P. Gupta, R. Janardan, J. Majhi and T. C. Woo, "Efficient geometric algorithms for workpiece orientation in 4- and 5-axis NC machining", Computer-Aided Design, 28(8), pp. 577–587, 1996.
17. W. Yang, H. Ding and Y. Xiong, "Manufacturability analysis for a sculptured surface using visibility cone computation", International Journal of Advanced Manufacturing Technology, 15, pp. 317–321, 1999.
18. G. Elber, "Accessibility in 5-axis milling environment", Computer-Aided Design, 26(11), pp. 796–802, 1994.
19. A. H. Nickel, "Analysis of thermal stress in shape deposition manufacturing of metal parts", PhD thesis, Stanford University, 1999.
20. J. H. Kao, "Process planning for additive/subtractive sold freeform fabrication using medial axis transition", PhD thesis, Stanford University, 1999.
21. K. Ramaswami, "Process planning for shape deposition manufacturing", PhD. thesis, Stanford University, 1997.
22. J. W. H. Tangelder and J. S. M. Vergeest, "Robust NC path generation for rapid shape prototyping", Journal of Design and Manufacturing, 4, pp. 281–292, 1994.
23. J. W. H. Tangelder, J. S. M. Vergeest and M. H. Overmars, "Interference-free NC machining using spatial planning and Minkowski operations", Computer-Aided Design, 4(30), pp. 277–286, 1998.
24. P. Ng, W. Fang, B. Li, J. Zou and H. Gong, "Motion compensation for five-axis rapid prototyping system", Rapid Prototyping Journal, 4(2), pp. 68–76, 1998.
25. I. Horváth, J .S. Vergeest, J. J. Broek, Z. Rusák and B. Smit, "Tool profile and tool path calculation for free-form thick-layered fabrication", Computer-Aided Design, 30(14), pp. 1097–1110, 1998.
26. W. C. Tse and Y. H. Chen, "A robotic system for rapid prototyping", Proceedings of the IEEE International Conference on Robotics and Automation, New Mexico, USA, vol. 3, pp. 1815–1820, 1997.
27. Y. Song and Y. H. Chen, "Feature based robot machining for rapid prototyping", Journal of Engineering Manufacture, Proceedings Part B, 213(B5), pp. 451–459, 1999.
28. Y. H. Chen and Y. Song, "The development of a layer-based machining system", Computer-Aided Design, 33(4), pp. 331–342, 2001.
29. F. Xu, H. T. Loh and Y. S. Wong, "Considerations and selection of optimal orientation for different rapid prototyping system", Rapid Prototyping Journal, 5(2), pp. 54–60, 1999.