

# A robotic system for rapid prototyping

W.C. Tse, Y.H. Chen

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

## ABSTRACT

Reducing prototyping time is a good way to make the product development cycle shorter. This can be achieved in two ways: one is to develop new prototyping technologies like stereolithography apparatus (SLA), selective laser sintering (SLS) etc.; the other is to improve the principal existing technique which is CNC based method. In this paper, a robotic system for rapid prototyping which is an enhancement of CNC based method is presented. A robot arm holding a milling tool is used to machine the prototype of a solid model drawn in commercial CAD systems. The rough cut and finish cut NC tool path for the robot arm are generated automatically from the solid model of an object. Objects may have different kinds of surfaces e.g. planar surface, general quadratic surface, B-spline surface and compound surface. The proposed method is implemented on the AutoCAD platform. A number of produced prototypes have shown satisfactory results.

**Keywords:** Robotics, NC Machining, Rapid Prototyping, Solid Modelling

## 1. INTRODUCTION

Industry is becoming more and more competitive than before. In order to survive, companies have to make their product development cycle as short as possible. One good way to reduce the product development time is by reducing the time of product prototyping. By reducing the prototyping time, the turnaround time for conceptual designs is much shorter. In addition, form, fit and function tests can be performed much earlier in the design cycle with a physical prototype, thus costly engineering change in production stage can be minimized. Also the amount of lost production time, retooling, rework and scrap waste incurred in manufacturing is reduced.

There are two approaches when trying to minimize the prototyping time, one is to develop new prototyping technology and the other is to improve the traditional prototyping methods. Rapid prototyping

technologies like Stereolithography apparatus (SLA), solid ground curing (SGC), selective laser sintering (SLS), laminated object manufacturing (LOM), fused deposition modelling (FDM), three-dimensional printing, ballistic particle manufacturing (BPM), shape-melting technology, photochemical machining and light sculpting are the results of the first approach generated during the last ten years. The second approach is to improve the CNC based machining method, which is the most powerful one among the traditional methods, in different aspects (e.g. domain of geometric model that can be handled, machining speed, size of prototype produced etc.). The second approach is taken in this research.

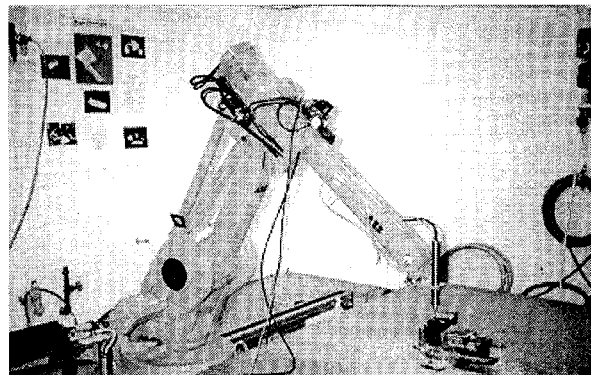


Figure 1. The robotic prototyping system

## 2. ROBOTIC PROTOTYPING SYSTEM

A robotic prototyping system is used instead of traditional CNC devices due to the following reasons: With the same size of workspace, the robotic prototyping system has a 40% of floor space reduction over ordinary CNC devices as stated in [1].

Due to the articulated robot configuration, it is more flexible in terms of tool access. Therefore it can handle more complicated geometric parts.

An overview of the robotic prototyping system is shown in Fig.1. A precision spindle is mounted on the end-effector of an articulated ABB IR1400 robot arm. The robot arm is in turn mounted on a track to provide

additional movement to the robot arm so that larger parts can be handled.

The solid model of the object to be prototyped is first created by the user in a computer aided design system. The user can then specify the type and diameter of the cutter and the desired accuracy of the prototype. A path file will then be generated automatically from the solid model of the object. It is used to control the robot arm to mill the stock material into the final prototype. The coordinate system of the machining process is shown in Fig.2. Since the research is still in a preliminary state, the milling tool axis is kept parallel to z-axis throughout the milling operation to simplify the algorithms. The stock is milled in a zig-zag fashion in two stages, rough cut and finish cut. Each stage has one or more layers.

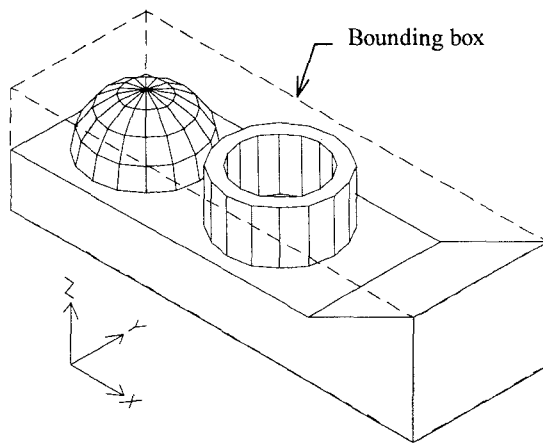


Figure 2. Bounding box of the object and coordinate system of the machining process

## 2. HEIGHT ARRAY

Each of the elements of the machining environment including object, milling tool, tool-holder and stock-in-progress are approximated by rectangular boxes and the height of each rectangular box is stored in a height array.

The generation of Object height array  $O[i][j]$  is used as an example to illustrate the construction process of a height array.  $O[i][j]$  is used to represent the object to be milled. Users must specify the solid model of the object and the desired machining accuracy. The first step of the construction process is to determine the bounding box of the object (Fig.2). Then the bounding box is divided into grids along the x-y plane according to the accuracy specified by the user (Fig.3). After that, a ray is projected onto the object along the z-direction through the centre of each grid and all the intersection points between this ray and the object are determined. The height of the rectangular box at the grid is then taken as

the highest z-value of the intersection points. If no intersection point is found, the height of the rectangular box is taken as 0. The rectangular box approximation of the object in Fig.2 is shown in Fig.4. The indexing method of  $O[i][j]$  is shown in Fig.3.

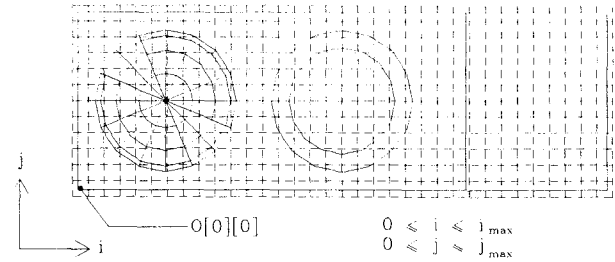


Figure 3. Grid cell generation and indexing method for  $O[i][j]$

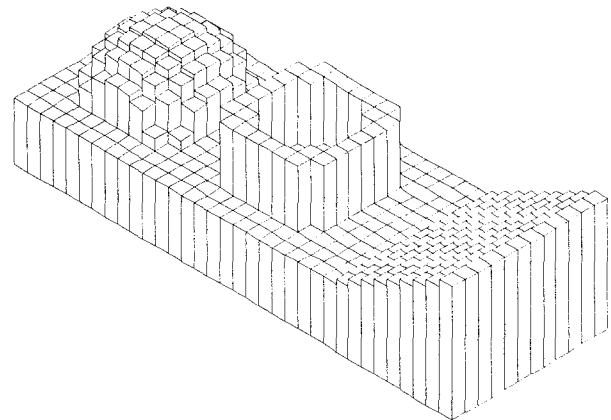


Figure 4. The rectangular box approximation of the object

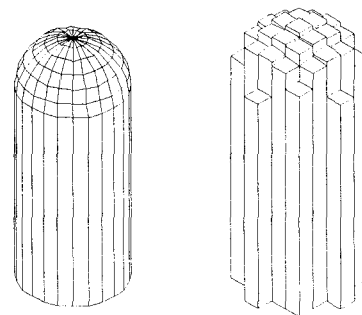


Figure 5. The solid model and the corresponding rectangular box approximation of a ball-end cutter

The Tool height array  $T[k][l]$  is used to represent the milling part of the cutter. The solid model of the tool is created automatically after the user has specified the type (ball-end or flat-end) and the diameter of the tool. Based on the solid model,  $T[k][l]$  is then generated automatically. A cylinder together with a half sphere are used to model a ball-end mill and a single

cylinder is used to model a flat-end mill. Fig.5 shows the solid model and the rectangular box approximation of a ball-end mill. Fig.6 shows the indexing method of  $T[k][l]$ .

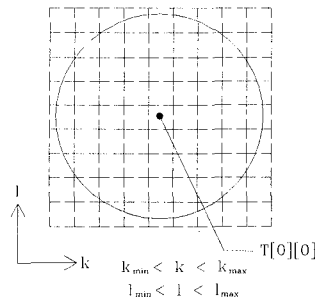


Figure 6. The indexing method of  $T[k][l]$

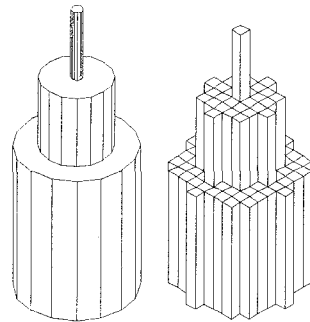


Figure 7. The solid model and the corresponding rectangular box approximation of the tool-holder

The tool-Holder height array  $H[k][l]$  is used to represent the non-milling part of cutter and the head of spindle. Three cylinders are used to model the tool-holder. The two cylinders at the bottom represent part of the spindle head and the top one is the non-milling part of tool.  $H[k][l]$  is produced based on this model. The tool-holder model and the rectangular box approximation are shown in Fig.7. The indexing method of  $H[k][l]$  is the same as that of the tool.

The Stock-in-progress height array  $S[i][j]$  is used to represent the stock-in-progress. It is updated during the machining operation to reflect change to the stock material. The indexing of  $S[i][j]$  is the same as that of the object.

### 3. ACCURACY CONSIDERATION

The grid size (resolution) for encoding the solid models (object, tool and tool-holder) into the corresponding rectangular box approximation should be selected such that the prototype produced can meet the accuracy requirement specified by user. The accuracy of the prototype with respect to the object is affected by

two factors: 1. Accuracy of rectangular box approximation with respect to object; 2. Accuracy of prototype with respect to rectangular box approximation.

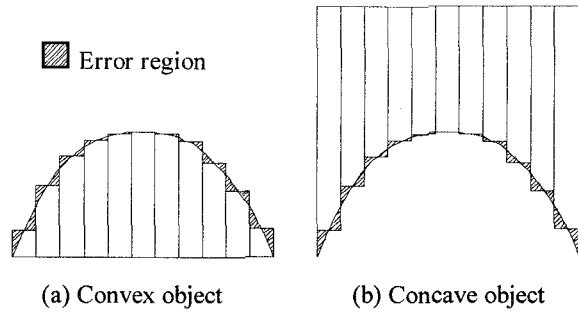


Figure 8. The error analysis

#### 3.1. Accuracy of rectangular box approximation with respect to object

In this section, the accuracy of the rectangular box approximation with respect to the original object will be discussed. Two measures namely curvature and “fatness” of a solid model will influence the error of the encoding process.

##### 3.1.1. Object with smooth boundary

As stated and proved by Stifier in [2], given the maximal curvature of a convex object  $\kappa$  and the maximum error of encoding process  $\alpha_1$ , the grid size  $d$  should be chosen to be smaller or equal to  $\min[\sqrt{2}/\kappa, \alpha_1/\sqrt{2}]$ , i.e.  $d \leq \min[\sqrt{2}/\kappa, \alpha_1/\sqrt{2}]$ , in order to meet accuracy requirement. The above accuracy criterion can be extended easily to concave object as follows — Without loss of generality, a 2-D case would be considered here. First consider a rectangular box approximation which has been encoded from a convex object with  $d \leq \min[\sqrt{2}/\kappa, \alpha_1/\sqrt{2}]$  (Fig.8(a)). The maximum distance from a point of the convex object to the rectangular box approximation is less than  $\alpha_1$ . Next we consider a concave object which is the “complement” of the above convex object (Fig.8(b)). It can be seen that the “error regions” of the concave object is just the same as that of the convex one. Therefore the rectangular box approximation of the concave object also has a maximum error of  $\alpha_1$  with respect to the original concave object.

##### 3.1.2. Object with non-smooth boundary

For object with non-smooth boundary, the maximal curvature is infinite. Thus the above measure

cannot be used to constrain the encoding error. The criterion for treating accuracy for non-smooth object is related to the extension of the object. Long and thin object is more difficult to model accurately than “fat” object such as a cube. The notion of “fat” object that has been introduced by [3] is used as a measure to ensure encoding accuracy for non-smooth object. Roughly speaking, fat object is an object for which the “width-to-length” ratio is greater than a given “fatness” constant. By definition, fat objects are always convex, however the same idea can be extended to non-convex object easily.

### 3.2. Accuracy of prototype with respect to object height array

Assume no tool interference and tool-holder collision occur, when the rectangular box approximation of tool instead of the actual tool is used to mill the stock, the rectangular box approximation of object should be what is left behind after the milling process. However, since the milling process is performed with the actual tool, there is a difference between the prototype produced and the rectangular box approximation of object. The maximum error of the prototype with respect to the rectangular box approximation of object is just the maximum error of the actual tool with respect to the rectangular box approximation of tool which is also  $\alpha_1$ .

### 3.3. Grid size

The error of the prototype produced with respect to original object would be the sum of the above two errors which is equal to  $2\alpha_1$ . The maximum error  $\alpha$  specified by user should be greater or equal to  $2\alpha_1$ , i.e.  $\alpha \geq 2\alpha_1$ . Thus, the finish cut grid size  $d_f$  should satisfy the condition,  $d_f \leq \min[\frac{\sqrt{2}}{\kappa}, \frac{\alpha}{2\sqrt{2}}]$ , in order to meet the accuracy requirement specified by the user.

## 4. PROTOTYPE CUTTING

The milling process includes two stages, rough cut and finish cut. Different milling stages have different functions. Rough cut is used to remove the surplus volume of stock material as fast as possible without the occurrence of overcutting. The function of finish cut is to get a good surface finish and precise dimensions of the prototype.

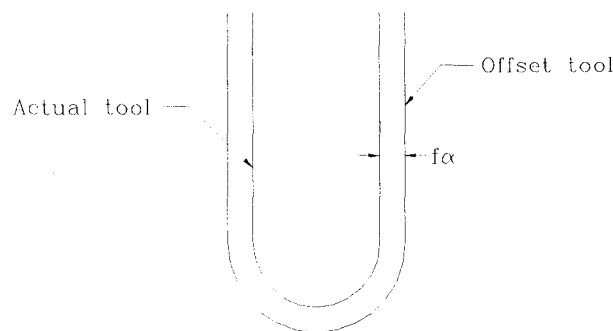


Figure 9. The offset tool and the actual tool

### 4.1. Rough cut

In order to increase the cutting speed and reduce the computation time, the rough cut grid size  $d_r$  used to generate the rectangular box approximations is  $f$  times of the finish cut one. The ratio  $f$  is called rough to finish factor which should be an odd integer greater than 1. Since  $d_r$  is larger than  $d_f$ , the error of the prototype produced by the rough cut process with respect to object is not  $\pm \alpha$  but  $\pm f\alpha$  (Assume  $d_r \leq \frac{\sqrt{2}}{\kappa}$ ). Overcutting of the object can be prevented by using a virtual tool model instead of the actual tool model to generate the rough cut tool height array  $T[k][l]$ . The virtual tool model is an “offset version” of the actual tool model by the amount of  $f\alpha$  as shown in Fig.9. The rough cut object height array  $O[i][j]$  cannot be used to generate the cutter location points directly since there may be tool interference and collision occurring during machining operation. Also, a virtual tool-holder model with an offset of  $f\alpha$  is used instead of the actual tool-holder model in order to avoid any possible collision with the actual stock.

### 4.2. Tool interference avoidance

In this context, tool interference means the overcutting of the rectangular box approximation of object when the tool try to achieve the height value of some grids. The maximum interference at each grid of the  $O[i][j]$  is determined and the tool is lifted up by this amount to avoid interference. This interference avoidance algorithm is similar to that used by [5].

### 4.3. Collision avoidance

In the machining operation, collisions between different elements may occur, e.g. collision between tool-holder and stock-in-progress, between tool-holder and fixture, between robot manipulator and mounting table etc. Since the research is now in a preliminary stage, so the size of prototype is limited, collisions other

than that between tool-holder and the stock-in-progress are not possible to occur. So we only avoid the collision between the tool-holder and the stock-in-progress. The collision avoidance problem is handled similarly to the interference avoidance problem. The maximum collision at each grid of the  $S[i][j]$  is determined and the tool-holder is lifted up by this amount to avoid collision.

#### 4.4. Finish cut

Accuracy and surface finish are the main factors that we concern most in finish cut. The grid size is calculated automatically according to the criterion stated in Section 3. Tool interference avoidance and collision avoidance are handled in the same way as in rough cut. However, for finish cut, the actual tool model is used for the generation of  $T[k][l]$ , so that the prototype produced has an error of  $\pm \alpha$  with respect to the object except where the tool is inaccessible due to interference or collision. In this case the virtual tool-holder model is the offset from the actual tool-holder model by  $\alpha$  in order to prevent collision.

#### 4.5. Stock-in-progress conversion

Since the grid size used in rough cut is different from that of finish cut, the stock-in-progress height array  $S[i][j]$  resulted from the rough cut process cannot be used directly for the finish cut calculation. We need a way to convert the rough cut  $S[i][j]$  (Fig.10(a)) to the finish cut one. The first step is to expand the rough cut  $S[i][j]$  into the same dimension as the finish cut  $S[i][j]$  as shown in Fig.10(b). For correct finish cut collision avoidance calculation, the maximum positive error of the actual stock with respect to the  $S[i][j]$  should be  $+\alpha/2$ , while in rough cut calculation, it is  $+f\alpha/2$ . The maximum positive error of the actual stock with respect to "expanded array" should be reduced to  $+\alpha/2$  before it can be used for finish cut collision avoidance calculation. The maximum positive error of the stock with respect to the rough cut  $S[i][j]$  is  $(f\alpha/2 + f\alpha) = 3f\alpha/2$  (The first part  $f\alpha/2$  is the error of the actual stock with respect to the rough cut  $S[i][j]$  if the actual tool is used in the rough cut process. The second part  $f\alpha$  is the amount of undercutting due to virtual tool.). This reduction can be achieved by offsetting the "expanded array" by an amount of  $\alpha/2*(3f - 1)$ . The offsetting process is done in two stages: The "expanded array" is first offset vertically as shown in Fig.11. Then it is offset horizontally by the smallest number of grid cell that can cover the value of  $\alpha/2*(3f - 1)$  as shown in Fig.12. Then this array can be used as the  $S[i][j]$  for the finish cut process.

60	55	70	40
41	68	80	37

(a) Rough cut  $S[i][j]$

60	60	60	55	55	55	70	70	70	40	40	40
60	60	60	55	55	55	70	70	70	40	40	40
60	60	60	55	55	55	70	70	70	40	40	40
41	41	41	68	68	68	80	80	80	37	37	37
41	41	41	68	68	68	80	80	80	37	37	37
41	41	41	68	68	68	80	80	80	37	37	37

(b) Expanded array with the same dimension as the finish cut  $S[i][j]$

Figure 10. The expansion of rough cut array  $S[i][j]$  ( $f=3$ )

62	62	62	57	57	57	72	72	72	42	42	42
62	62	62	57	57	57	72	72	72	42	42	42
62	62	62	57	57	57	72	72	72	42	42	42
43	43	43	70	70	70	82	82	82	39	39	39
43	43	43	70	70	70	82	82	82	39	39	39
43	43	43	70	70	70	82	82	82	39	39	39

Figure 11. The "expanded array" after vertical offsetting of 2

62	62	62	62	57	72	72	72	72	72	42	42
62	62	62	62	57	72	72	72	72	72	42	42
62	62	70	70	70	82	82	82	82	82	42	42
62	62	70	70	70	82	82	82	82	82	42	42
43	43	70	70	70	82	82	82	82	82	39	39
43	43	70	70	70	82	82	82	82	82	39	39

Figure 12. The "expanded array" after horizontal offsetting of one grid cell (grid size : 2.5, offset amount: 2)

## 5. RESULTS

The algorithms have been implemented on the AutoCAD platform, a prototype for the object shown in Fig.2 has been produced by an ABB IR1400 robot as a sample. Fig.13 show the tool path of the last rough cut layer of the prototype. The rough cut process is shown in Fig.14 and the prototype produced after the finish cut process is shown in Fig.15.

## 6. CONCLUSION

An automatic tool path generation method using height array has been presented. It can generate tool paths for both rough cut and finish cut with different grid size. When implemented in a robotic machining system, satisfactory prototypes can be produced. The accuracy of the prototype with respect to the geometric model has also been discussed.

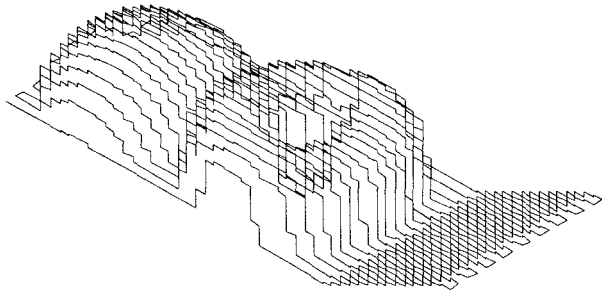


Figure 13. The last cutting layer of the rough cut process

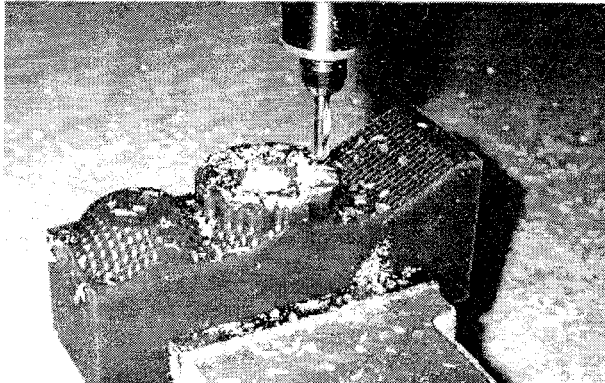


Figure 14. The rough cut process

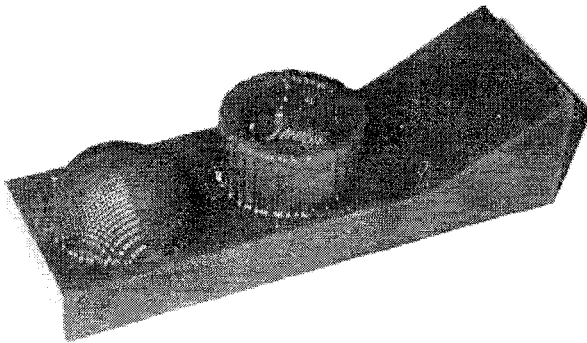


Figure 15. The prototype produced after the finish cut process

## 7. REFERENCES

- [1] Belforte, D.A., "Robotic manipulation for laser processing." Proceedings of the SPIE-High power lasers and their industrial applications. Vol. 650(1986), 262-270.
- [2] Sabine Stifter, "Simulation of NC machining based on the dixel model: a critical analysis", Int J Adv Manuf Technol 10 (1995), P.149-157.
- [3] J. Nievergelt and P. Widmayer, "Guard algorithms answer stabbing and intersection queries for fat spatial object", Dagstuhl-Seminar-Report 22, H. Alt, B. Chazelle and E. Weizi (ed.), Computational Geometry, October 1991.
- [4] Takafumi Saito and Tokiichiro Takahashi, "NC machining with G-buffer method.", Computer Graphics, Volume 25 (1991), Number 4, P 207-216.
- [5] Chun-Fong You and Chih-Hsing Chu, "An automatic path generation method of NC rough cut machining from solid models.", Computer in Industry 26 (1995), P 161-173.
- [6] Johan W.H. Tangelder and Joris S.M. Vergeest, "Robust NC path generation for rapid shape prototyping.", Journal of Design and Manufacturing 4 (1994), P281-292.