Preprocessor with spline interpolation for converting stereolithography into cutter location source data

Fusaomi Nagata[1, a], Yudai Okada[1], Tatsuhiko Sakamoto[1], Takamasa Kusano[2, b], Maki K. Habib[3, c], Keigo Watanabe[4, d]

[1]Department of Mechanical Engineering, Faculty of Engineering, Tokyo University of Science, Yamaguchi, 1-1-1 Daigaku-Dori, Sanyo-Onoda 756-0884, Japan
[2]SOLIC Co., Ltd. 80-42 Yotsuyama-Machi, Omuta 836-0067, Japan
[3]American University in Cairo, AUC Avenue, P.O.Box 74, New Cairo 11835, Egypt
[4]Graduate School of Natural Science and Technology, Okayama University, 3-1-1 Tsushima-naka, Kitaku, Okayama 700-8530, Japan
Email: [a]nagata@rs.tusy.ac.jp, [b]solic@sweet.ocn.ne.jp, [c]maki@aucegypt.edu, [d]watanabe@sys.okayama-u.ac.jp

ABSTRACT. The authors have developed earlier an industrial machining robotic system for foamed polystyrene materials. The developed robotic CAM system provided a simple and effective interface without the need to use any robot language between operators and the machining robot. In this paper, a preprocessor for generating Cutter Location Source data (CLS data) from Stereolithography (STL data) is first proposed for robotic machining. The preprocessor enables to control the machining robot directly using STL data without using any commercially provided CAM system. The STL deals with a triangular representation for a curved surface geometry. The preprocessor allows machining robots to be controlled through a zigzag or spiral path directly calculated from STL data. Then, a smart spline interpolation method is proposed and implemented for smoothing coarse CLS data. The effectiveness and potential of the developed approaches are demonstrated through experiments on actual machining and interpolation.

1. Introduction

In manufacturing industries, there exist two representative systems for prototyping. One is the conventional removal manufacturing systems, such as NC milling or NC lathe machines which can precisely perform metalworking and woodworking. The other is the additive manufacturing systems, such as optical shaping apparatus or 3D printer which enables to quickly transform a design concept into a real model. As for removal machining, Lee introduced a machining automation using an industrial robot [1]. The robot had double parallel mechanism and consequently performed a large work space as well as a high stiffness to reduce deformation and vibration. Schreck et al. launched Hard Material Small-Batch Industrial Machining Robot (HEPHESTOS) project, where the focus of the objective was on developing robotic manufacturing methods in order to give rise to a cost-efficient solution in hard materials machining [2].

As one of examples of post-processor in CAM system, CLS data written in ISO format produced by main-processor in CAM system could be transformed into G-codes files (Numerical Control files) and an industrial 5-axis machine with a nutating table could be actually controlled using the NC files [3].

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

However, it is not easy but complicated for the CAM system to generate each robot language according to different industrial robot makers.

The authors developed an industrial machining robotic system for foamed polystyrene materials as shown in Fig. 1(a) [4]. In the machining robot, the developed robotic CAM system called the direct servo system provided a simple interface for NC data and CLS ones, without the need to use any robot language between operators and the machining robot [5]. In other words, a desired trajectory consisting of numerous discrete position and orientation components can be calculated every sampling period. The direct servo system simplified the machining process without a post-processor as shown in Fig. 1(a). However, a CAM process to generate CLS data after the design process has to be further passed through in order to machine the designed model using the robot. Although the authors surveyed related papers, e.g., [6], [7] to make the process easier, any suitable system was not seen.

In this paper, a robotic preprocessor is first proposed for the machining robot to directly convert STL data into CLS data as illustrated by the proposed process 2 shown in Fig. 1(b), and this helps to remove the need for having the conventional CAM process. The STL originally means Stereolithography which is a file format proposed by 3D Systems and recently is supported by many CAD/CAM software. It is also known as Standard Triangulated Language in Japan. The STL is widely used for rapid prototyping with a 3D printer which is a typical additive manufacturing system [8]. The STL deals with a triangular representation of a 3D surface geometry [9], [10]. The robotic preprocessor allows the machining robot to be controlled along zigzag paths or spiral ones generated based on STL data. Then, a smart spline interpolation method, which is easily implemented, is proposed for smoothing coarse CLS data. The effectiveness and potential of this unique machining system with a smart spline interpolation are demonstrated through machining experiments of a foamed polystyrene and a wood.

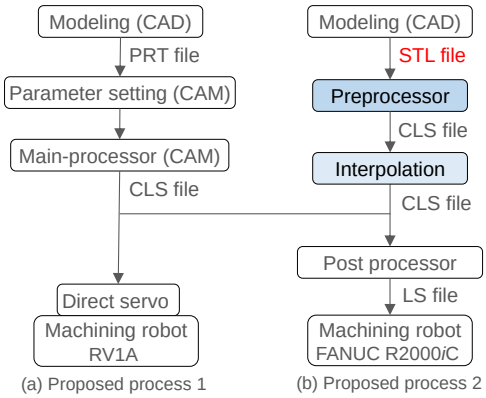


Figure 1. Comparing the proposed two processes for robotic machining.

2. Machining robot with robotic CAM system

A robotic post processor shown in Fig. 1(b) was already proposed to enhance the affinity between FANUC industrial robots and 3D CAD/CAM systems [12]. Figure 2 shows the machining scene of an impeller model using FANUC industrial robot R2000*i*C whose robot language is called LS. A ball-end mill attached to the arm tip can be controlled as to follow position and orientation components in LS data. The post processor is able to generate LS data directly from CLS ones. Our current interest is to enable the industrial machining robot to run through STL data that consists of numerous unstructured triangulated patches as shown in Fig. 3. $\mathbf{v}_{1i}$, $\mathbf{v}_{2i}$, $\mathbf{v}_{3i}$ and $\mathbf{n}_i$ are the three position vectors and a normal vector in *i*th triangular patch, respectively. In this paper, a preprocessor is proposed to convert STL data into two kinds of CLS data and it is integrated with the developed industrial machining robotic system

to execute an assigned machining job using CLS data. Accordingly, the system can implement its task and control its sequence of machining actions based on STL data.



Figure 2. Machining robot based on FANUC R2000*i*C.

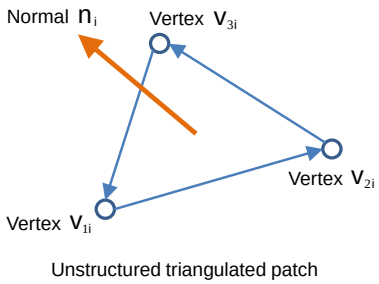


Figure 3. Unstructured triangulated patch in STL file.

3. Preprocessor by analyzing triangle patches in STL DATA

3.1 Automatic dimension extraction of STL data

In the early developed system, CLS data was basically generated along the original STL data consisting of many triangle patches [11]. In this subsection, the paper introduces an advanced process, in which the CLS data for robot control are generated along a zigzag path or spiral one by intelligently analyzing STL data. The significant advantage of this approach is to eliminate the need to use any commercial CAM software, and accordingly, 3D printer-like data interface can be smartly realized. First of all, dimensions of the STL data are extracted by retrieving all patches in the STL file and they are set to two constants $\mathbf{v}_{min} = [x_{min}\ y_{min}\ z_{min}]^T$ and $\mathbf{v}_{max} = [x_{max}\ y_{max}\ z_{max}]^T$. From the next subsection, two types of base paths viewed in *xy*-plane are designed considering the extracted dimensions. They are a zigzag path and a spiral one.

3.2 Design of base zigzag path

This subsection explains how a base zigzag path is designed. Two effective machining parameters, i.e., pick feed and step, are respectively set to p_f and s_p by referring the dimensions. s_p is a constant pitch viewed in *xy*-plane between two adjacent points $\mathbf{c}_j = [c_{xj}\ c_{yj}\ 0]^T$ and $\mathbf{c}_{j+1} = [c_{x(j+1)}\ c_{y(j+1)}\ 0]^T$ on a zigzag path. One pass point $\mathbf{c}_j$ is appended into a CLS file as a "GOTO" statement. $j\ (1 \le j \le m)$ is the number of the pass points generated from STL data. m is the total number of triangle patches in the STL file. Figure 4 illustrates an example of a base zigzag path drawn within STL data consisting of multiple triangulated patches. As can be seen, c_{xj} and c_{yj} are located just on the zigzag path, so that remained height c_{zj} has only to be determined by analyzing triangle patches in the STL data.

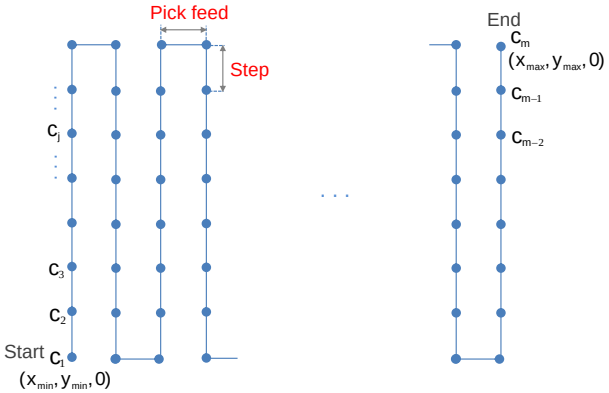


Figure 4. Base zigzag path $\boldsymbol{c}_j = [c_{xj}\, c_{yj}\, 0]^T$ along STL data consisting of multiple triangulated patches.

3.3 Generation of CLS Data along Base Zigzag Path

Figure 5 shows an example of generation of pass points $\mathbf{c}_j = [c_{xj}\ c_{yj}\ c_{zj}]^T$ in a patch (2). The number of pass points within the patch depends on the length of the step illustrated in Fig. 4. The dashed line in the upper figure shows one section of the base zigzag paths $\mathbf{c}_j = [c_{xj}\ c_{yj}\ 0]^T$ viewed in *xy*-plane and the chained line in the lower figure draws the generated path $\mathbf{c}_j = [c_{xj}\ c_{yj}\ c_{zj}]^T$ along the triangulated patch viewed in *yz*-plane. The pass points for constructing CLS data are generated along the chained line. The pass points in other patches such as (1), (3), (4) and (5) can be similarly obtained.

To realize the preprocessor based on STL data without using any CAM system, the *z*-component c_{zj} must be calculated just on a triangulated patch. In order to calculate c_{zj}, first of all, a triangulated patch, in which the point $\mathbf{c}_j$ viewed in *xy*-plane is included, is searched in the target STL data. Figure 6 shows the scene where the pass point $\mathbf{c}_j$ is located within a patch. Note that $\mathbf{c}_{j-1}$ and $\mathbf{c}_{j+1}$ may be also within the triangle patch as shown in Fig. 5. Whether $\mathbf{c}_j$ is located in the patch or not can be known by checking the following outer products.

$$(v_{2i} - v_{1i}) \times (c_j - v_{1i}) \quad (1)$$

$$(v_{3i} - v_{2i}) \times (c_j - v_{2i}) \quad (2)$$

$$(v_{1i} - v_{3i}) \times (c_j - v_{3i}) \quad (3)$$

If $\mathbf{c}_j$ is located within the patch, then the above three equations have the same sign. After finding the first triangle satisfying this condition, the equation of the plane including the triangle is determined by the perpendicular condition to the normal vector $\mathbf{n}_i$, which leads to

$$n_{xi}(x - x_{1i}) + n_{yi}(y - y_{1i}) + n_{zi}(z - z_{1i}) = 0 \quad (4)$$

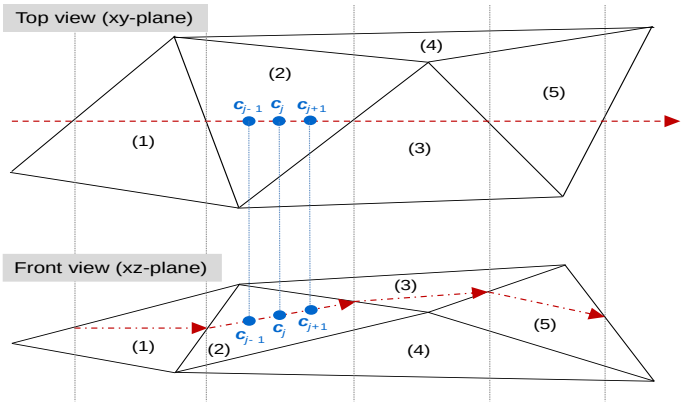


Figure 5. Generation of pass points. Upper and lower figures show the top and front views of five adjacent triangulated patches, respectively.

By respectively substituting c_{xj} and c_{yj} into x and y in Eq. (4), if $n_{zi} \neq 0$ then z-directional component c_{zj} can be calculated by

$$c_{zj} = z_{1i} - \frac{1}{n_{zi}}\{n_{xi}(c_{xj} - x_{1i}) + n_{yi}(c_{yj} - y_{1i})\} \quad (5)$$

where c_{xj} and c_{yj} are extracted from the base path shown in Fig. 4. Consequently, by repeating the above calculations, all pass points $\mathbf{c}_j = [c_{xj}\ c_{yj}\ c_{zj}]^T (1 \le j \le m)$ can be obtained. Figure 7 shows an example of regular and precise zigzag path (CLS data) generated from a STL file by the preprocessor.

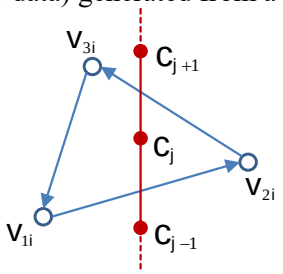


Figure 6. Pass point $\mathbf{c}_j$ is located within a triangle patch viewed in xy-plane.

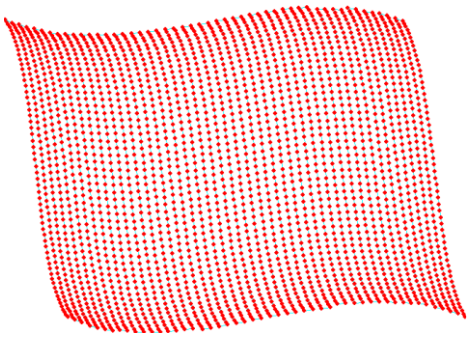


Figure 7. An example of regular and precise zigzag path along curved surface generated by the preprocessor.

3.4 Generation of CLS data along base circular spiral path

The base spiral $[c_{xj}\ c_{yj}\ 0]^T$ viewed in xy-plane for generating CLS data $[c_{xj}\ c_{yj}\ c_{zj}]^T$ is designed by

$$c_{xj} = \frac{r\theta_j}{2\pi}\cos\theta_j \quad (6)$$

$$c_{yj} = \frac{r\theta_j}{2\pi}\sin\theta_j \quad (7)$$

$$\frac{dc_{xj}}{d\theta_j} = \frac{r}{2\pi}(\cos\theta_j - \theta_j\sin\theta_j) \quad (8)$$

$$\frac{dc_{yj}}{d\theta_j} = \frac{r}{2\pi}(\sin\theta_j + \theta_j\cos\theta_j) \quad (9)$$

When pass points $[c_{xj}\ c_{yj}\ 0]^T$ viewed in xy-plane are generated while drawing a spiral path, two effective parameters need to be considered as shown in Fig. 8. One is the movement l_j [mm] from a pass point $\mathbf{c}_{j-1}$ to next one $\mathbf{c}_j$ along the spiral, i.e., arc length. The other is the pitch r [mm] along the radius vector. Operators can design preferred base spiral path only by giving l_j and r.

The rate between the length l_j of an arc on the spiral and the varying angle θ_j is calculated by

$$\frac{dl_j}{d\theta_j} = \sqrt{\left(\frac{dc_{xj}}{d\theta_j}\right)^2 + \left(\frac{dc_{yj}}{d\theta_j}\right)^2} = \frac{r}{2\pi}\sqrt{1+\theta_j^2} \quad (10)$$

To produce better surface quality with less cusp heights in actual removal machining using a ball-end mill, l_j shown in Fig. 8 must be a constant value l. Hence, the angle θ_j needs to be varied as

$$\theta_j = \theta_{j-1} + \Delta\theta_j \quad (11)$$

$$\Delta\theta_j = \frac{l}{r}\frac{2\pi}{\sqrt{1+\theta_{j-1}^2}} \quad (12)$$

This means $_j$ has only to be adjusted by Eqs. (11) and (12) in order that l_j is fixed to a constant value l. Figure 9 shows a result of circular spiral path (CLS data) generated by the preprocessor viewed from four angles, in which l and r are set to 1 mm. Note that c_{zj} can be calculated by Eq. (5).

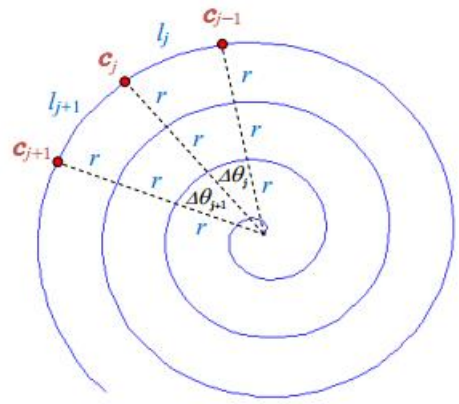


Figure 8. Base spiral path $\boldsymbol{c}_j = [c_{xj}\ c_{yj}\ 0]^T$ for generating CLS.

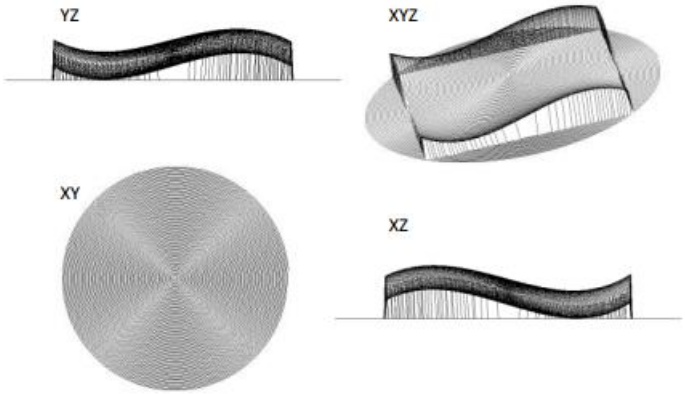


Figure 9. Regular and precise circular spiral path (CLS data) $\boldsymbol{c}_j = [c_{xj}\ c_{yj}\ c_{zj}]^T$ viewed in *xy*-plane generated by the preprocessor.

3.5 Removal machining experiment

In the earlier subsections, the preprocessor that can generate two kinds of regular and precise tool paths has been proposed. In this subsection, two machining experiments were conducted using a machining robot and a desktop NC machine tool, in which two tool paths (CLS data) shown in Figs. 7 and 11 generated by the preprocessor are given, respectively. Figure 10 shows the successful machining scene of the machining robot using the CLS data made along a zigzag path (left photo) and the resultant surface (right photo). The material is a foamed polystyrene. Also, Figs. 11 and 12 show another example of conversion from STL to spiral-based CLS data and its machining scene of desktop NC machine tool, respectively. The feasibility and effectiveness were confirmed from the actual experiments using the machining robot and the desktop NC machine tool. The material is a chemical wood.

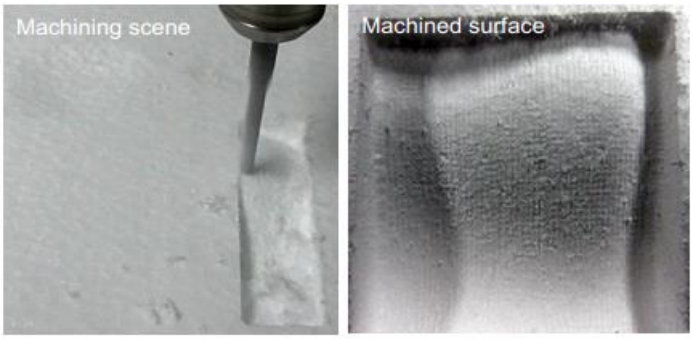


Figure 10. Machining scene of machining robot RV1A using zigzag-based CLS data shown in Fig. 7 (left); resultant surface (right).

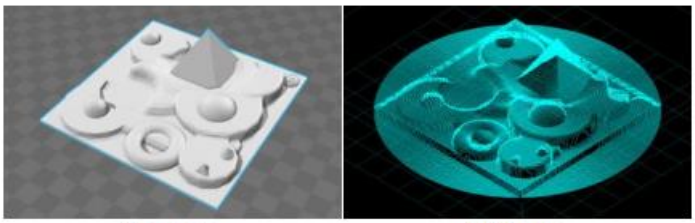


Figure 11. Another example of conversion from STL to spiral-based CLS data using the preprocessor.



Figure 12. Machining scene using a desktop NC machine tool, where the spiral-based CLS data shown in Fig. 11 generated by preprocessor are given.

4. Smart spline interpoation

When the density of CLS data is not high but coarse, some interpolation method is effective to smoothly reconstruct the trajectory with preferable scaling. Figure 11 illustrates a third-order spline curve consisting of three cubic curves $\boldsymbol{x}_{i-1}(t)$, $\boldsymbol{x}_i(t)$ and $\boldsymbol{x}_{i+1}(t)$ just passing on four points $\boldsymbol{p}_{i-1}$, $\boldsymbol{p}_i$, $\boldsymbol{p}_{i+1}$, $\boldsymbol{p}_{i+2}$ ($\in \Re^{3\times1}$). The 3rd order spline curve $\boldsymbol{x}_i(t)=[x_i(t)\ y_i(t)\ z_i(t)\ ]^T$ in Cartesian space is written as

$$\boldsymbol{x}_i(t)=\boldsymbol{a}_i t^3+\boldsymbol{b}_i t^2+\boldsymbol{c}_i t+\boldsymbol{d}_i \qquad (13)$$

where $\boldsymbol{a}_i=[a_{xi}\ a_{yi}\ a_{zi}\ ]^T$, $\boldsymbol{b}_i=[b_{xi}\ b_{yi}\ b_{zi}\ ]^T$, $\boldsymbol{c}_i=[c_{xi}\ c_{yi}\ c_{zi}\ ]^T$, $\boldsymbol{d}_i=[d_{xi}\ d_{yi}\ d_{zi}\ ]^T$ are the coefficient vectors; t is the normalized variable. When the x-directional component is considered, the following relations are obtained from Fig. 13.

$$x_i(\quad 1)= p_{x(i\quad 1)} = \quad a_{xi} + b_{xi} \quad c_{xi} + d_{xi} \tag{14}$$

$$x_i(0)= p_{xi} =d_{xi} \tag{15}$$

$$x_i(1)= p_{x(i+1)} =a_{xi} + b_{xi} +c_{xi} + d_{xi} \tag{16}$$

$$x_i(2)= p_{x(i+2)} =8a_{xi} +4b_{xi} +2c_{xi} + d_{xi} \tag{17}$$

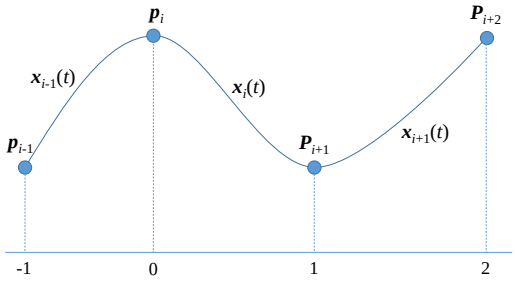


Figure 13. Spline interpolation using 3rd order functions.

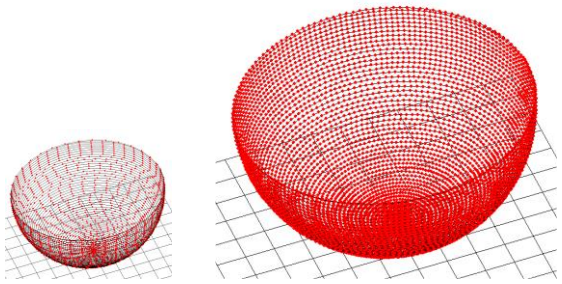


Figure 14. Original CLS data (left), scaled and interpolated CLS data (right) illustrated in perspective view.

d_{xi} is fixed as p_{xi}, so that the remaining coefficients can be calculated by

$$\begin{pmatrix} a_{xi} \\ b_{xi} \\ c_{xi} \end{pmatrix} = \begin{pmatrix} -1/6 & 1/2 & -1/6 \\ -1/2 & 1/2 & 0 \\ 1/3 & 1 & -1/6 \end{pmatrix} \begin{pmatrix} p_{x(i-1)} - p_{xi} \\ p_{x(i+1)} - p_{xi} \\ p_{x(i+2)} - p_{xi} \end{pmatrix} \tag{18}$$

Coefficients in other two directions are similarly obtained, so that the section between $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$ can be simply interpolated with $\boldsymbol{x}_i(t)$ $(0 \le t \le 1)$.

An experiment of scaling interpolation of a spiral path was conducted with the 3rd order spline curve. The left side of Fig. 14 shows the points in the original CLS data. Also, the right side of Fig. 14 draws the points in the CLS data interpolated with the 3rd-order spline curve, which is scaled up with 200%. The number of interpolated points within a section is set to three. As can been, the coarse original CLS data were desirably smoothed.

5. Conclusions

The STL file format was designed for fabbers in 1989. Fabbers means specialists who can perform 3D rapid prototyping from digital data, e.g., using a 3D printer. The STL means Stereolithography which is a file format proposed by 3D Systems and recently is supported by many CAD/CAM software. The 3D printer is currently recognized as a typical additive manufacturing system. In this paper, a robotic preprocessor has been proposed for a machining robot of foamed polystyrene to convert STL into CLS

data forming a zigzag or spiral path. The robotic preprocessor has allowed the machining robot to be controlled along a continuous zigzag or spiral path smartly calculated from STL data. Then, a 3rd –order spline interpolation method, which is easily implemented, has been proposed for smoothing originally coarse CLS data. The effectiveness and promise of this unique machining system are demonstrated through actual machining experiments and interpolation tests. The noteworthy point is that the proposed preprocessor has realized a promising data interface available with STL data like 3D printers and a smoother with variable scaling. Needless to say, the preprocessor can be applied to NC machine tools that can deal with CLS data.

Acknowledgments
This work was supported by JSPS KAKENHI Grant Number 25420232 and 16K06203.

References

[1] M.K. Lee, Design of a high stiffness machining robot arm using double parallel mechanisms, *in Procs. of 1995 IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 234–240, (1995).
[2] G. Schreck, D. Surdilovic and J. Krueger, HEPHESTOS: Hard material small-batch industrial machining robot, *in Procs. of 41st International Symposium on Robotics (ISR/Robotik 2014)*, pp. 1–6, (2014).
[3] C.A. My, Integration of CAM systems into multi-axes computerized numerical control machines, *in Procs. of 2010 Second International Conference on Knowledge and Systems Engineering (KSE)*, pp. 119–124, (2010).
[4] F. Nagata, A. Otsuka, K. Watanabe and M.K. Habib, Fuzzy feed rate controller for a machining robot, *in Procs. of the 2014 IEEE International Conference on Mechatronics and Automation (IEEE ICMA 2014)*, pp. 198–203, (2014).
[5] F. Nagata, S. Yoshitake, A. Otsuka, K. Watanabe and M. K. Habib, Development of CAM system based on industrial robotic servo controller without using robot language, *Robotics and Computer-Integrated Manufacturing*, Vol. 29, No. 2, pp. 454–462, (2013).
[6] A. Al-Ahmari and K. Moiduddin, CAD issues in additive manufacturing, *Comprehensive Materials Processing, Vol. 10: Advances in Additive Manufacturing and Tooling*, pp. 375–399, (2014).
[7] A.K. Matta, D.R. Raju and K.N.S. Suman, The integration of CAD/CAM and rapid prototyping in product development: a review, *Materials Today: Proceedings*, Vol. 2, Nos. 4/5, pp. 3438–3445, (2015).
[8] A.C. Brown and D.D. Beer, Development of a stereolithography (STL) slicing and G-code generation algorithm for an entry level 3-D printer, *in Procs. of IEEE African Conference 2013*, pp. 1–5, (2013).
[9] M. Szilvasi-Nagy and G. Matyasi, Analysis of STL files, *Mathematical and Computer Modeling*, Vol. 38, Nos. 7/9, pp. 945–960, (2003).
[10] C. Iancu, D. Iancu and A. Stancioiu, From CAD model to 3D print via STL file format, *Fiability & Durability*, Vol. 1, No. 5, pp. 73–81, (2010).
[11] F. Nagata, K. Takeshita, K. Watanabe and M.K. Habib, Generation of triangulated patches smoothed from original point cloud data with noise and its application to robotic machining, *in Procs. of The 2016 IEEE International Conference on Mechatronics and Automation (ICMA 2016)*, pp. 535–540, (2016).
[12] F. Nagata, Y. Yamane, Y. Okada, T. Kusano, K. Watanabe and M.K. Habib, Post processor for an industrial robot FANUC R2000iC, *in Procs. of 22nd International Symposium on Artificial Life and Robotics*, 5 pages, (2017).