International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:7, No:7, 2013

Efficient Web-Learning Collision Detection Tool on Five-Axis Machine

Chia-Jung Chen, Rong-Shine Lin, and Rong-Guey Chang

Abstract**—**As networking has become popular, Web-learning tends to be a trend while designing a tool. Moreover, five-axis machining has been widely used in industry recently; however, it has potential axial table colliding problems. Thus this paper aims at proposing an efficient web-learning collision detection tool on five-axis machining. However, collision detection consumes heavy resource that few devices can support, thus this research uses a systematic approach based on web knowledge to detect collision. The methodologies include the kinematics analyses for five-axis motions, separating axis method for collision detection, and computer simulation for verification. The machine structure is modeled as STL format in CAD software. The input to the detection system is the g-code part program, which describes the tool motions to produce the part surface. This research produced a simulation program with C programming language and demonstrated a five-axis machining example with collision detection on web site. The system simulates the five-axis CNC motion for tool trajectory and detects for any collisions according to the input g-codes and also supports high-performance web service benefiting from C. The result shows that our method improves 4.5 time of computational efficiency, comparing to the conventional detection method.

Keywords**—**Collision detection, Five-axis machining, Separating axis.

I. INTRODUCTION

THE Computer Numerically Controlled (CNC) machine tools have been widely used in industry. More advanced machining technologies uses five-axis CNC machining for complicated part fabrication. Five-axis machine has two rotary tables that improve the machining capability and quality. However, five-axis machining has high chances of axial table colliding due to the coordinated motion for rotating and translating tables. If the machine tool collides during machining, it damages the machine itself and delays the productions. Therefore, how to prevent the axial table collisions during machining is very important for five-axis machining processes.

Interference detection methods among objects have been presented by many researchers. Palmer proposed bounding sphere method that detected any interference between 2 objects very quickly by checking the distance of 2 bounding spheres [1]. Bradshaw used Octree data structure to improve the bounding sphere method that made the object closer to the approximated spheres [2]. Instead of using bounding sphere, Cohen et al. used bounding box for collision detection [3]. Bergen improved the bounding box method by partitioning the object into smaller areas [4]. Gottschalk et al. proposed oriented bounding box method that made the collision detection more precise [5]. Chang et al. applied separating axis theorem on oriented bounding box method to speed up the detecting process [6].

Nowadays, Web-learning tool [7] has been an important issue at CNC tool field because CNC engineers must be trained for a long time. Even for well-trained engineer, using real machine without collision detection maybe damage hardware during execution. Therefore, well-defined web tools can help engineer to execute g-code and manipulate a CNC machine. For reality, Web tools need the support of a 3D virtual machine tool to simulate real machines [8]. In this paper, we use OpenGL [9] to implement the proposed web tool. General speaking, heavy computing of 3D objects makes OpenGL difficult to be integrated with any device. Therefore, we present an efficient web tool to overcome this hurdle above for five-axis machining collision detection. This research uses a systematic approach for collision detection based on the high performance of C programming language. The methodologies include the kinematics analyses for five-axis motions, separating axis method for collision detection, computer simulation for detection verification, and the implementation to construct web services [10], [11]. The collision detection process needs to construct the five-axis machine, according to the real machine configuration. To implement the collision detection process, the machine is modeled in terms of STL format, a triangular facet approximation. The input to the detection system is the g-code part program, which describes the tool motions to produce the part surface. The system simulates the CNC motion for tool trajectory. The axial table location is detected for any collisions according to the g-codes, using separation axis method. The machining simulation will stop and show the collided g-code block if any collision is detected. The improved efficiency for computation time is analyzed with demonstrated example in the end of this paper.

II. FIVE-AXIS MACHINE TOOLS

The commonly used five-axis machine tools are table tilting type, spindle tilting type, and table/spindle tilting type. This research uses table tilting type five-axis machine for theorem derivation and explanation. However, the algorithm can be used

Chia-Jung Chen and Rong-Guey Chang are with the Advanced Institute of Manufacturing with High-tech Innovations and the Department of Computer Science and with the information Engineering, National Chung Cheng University, Chia-Yi, Taiwan 62102, R.O.C (e-mail: ccj98p@cs.ccu.edu.tw ,rgchang@cs.ccu.edu.te).

Rong-Shine Lin is with the Advanced Institute of Manufacturing with High-tech Innovations and the Department of Mechanical Engineering, National Chung Cheng University, Chia-Yi, Taiwan 62102, R.O.C. (e-mail: imelin@ccu.edu.tw).

for any types. A commercial available table tilting type five-axis machine is shown in Fig. 1.

Fig. 1 Five-Axis Machine

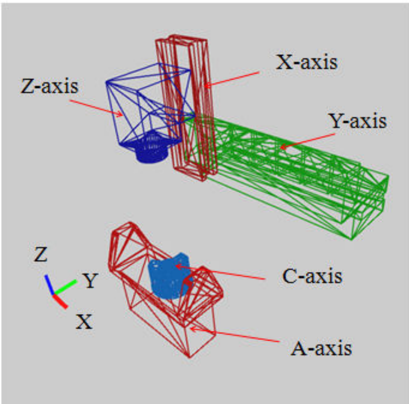

Fig. 2 STL Object

A. CAD model for Machine Tools

The machine design is usually modeled as STL format, which uses triangular facets to describe the designed object. Each triangular facet contains 3 vertices and a facial normal vector. Most of the commercial available Computer-Aided Design (CAD) softwares provide STL format for object design output. Fig. 2 shows the STL model for the machine shown in Fig. 1.

TABLE I
THE KINEMATICS RELATIONSHIP AMONG AXES

G-Code \ Moving axes	X axis	Y axis	Z axis	A axis	C axis
X	O		O		
Y	O	O	O		
Z			O		
A				O	O
C					O

B. Kinematics Analyses

The machine tool reads the g-code part program, which controls the cutting tool to produce the part. The typical 5-axis machining g-code example for linear motion is shown as, "G01 X100 Y200 Z300 A10 C20". This g-code moves X-axis, Y-axis, Z-axis, A-axis, and C-axis for 100, 200, 300, 10, and 20 units, respectively. The kinematics depends on the structure of a machine tool. Table I shows the kinematic relationship between axes for the machine structure shown in Fig. 1. The left column means reading g-codes for an axial motion that makes the other axial motion simultaneously. For example, if a g-code moves X axis, the Z-axis has to be moved simultaneous. Likewise, if A-axis moves, the C-axis has to be moved, too.

The typical collision detection process for a g-code motion on five-axis machine tools needs to detect 10 cases, X-to-Y, X-to-Z, X-to-A, X-to-C, Y-to-Z, Y-to-A, Y-to-C, Z-to-A, Z-to-C, and A-to-C. Although there are 10 detecting cases for five–axis motions, some detections can be waived if the axial structure is special arranged or the axial traveling distance is restricted. For the machine shown in Fig. 1, the required detections are z-axis table with respective to the A and C axes only.

C. Web Learning Tool with 3D Simulation

Due to high cost and accuracy of CNC machine tools, users usually pick up CNC knowledge via a long training. Without training, new users may spend much time and unnecessary cost to learn how to manipulate machine. In fact, even well-trained users sometimes manipulate machine in a wrong way. Therefore, web learning tool with 3D simulation is a useful tool for CNC users to remedy the issue above. However, 3D simulation usually results in intensive computation and thus leads to slow performance. In this paper, we present a design and implementation of an efficient web learning tool with 3D simulation based on OpenGL for collision detection.

III. OBJECT DETECTION ALGORITHM

A. Separating Axis Algorithm

Separating Axis Theorem for 2-D space states that two convex polygons do not intersect if and only if there exists a line such that the projections of the two polygons onto the line do not intersect. The line is called separating axis. In other words, it can be defined as two convex polygons do not intersect if and only if there exists a line that completely divides a polygon on one side of the line and the other polygon on the other side of the line. This line is called the separating line, which is perpendicular to the separating axis.

Separating Axis Theorem can apply to 3-D space for detecting if two objects interfered or not. Two objects do not intersect if and only if there exists a separating plane that completely separates these two objects. The separating plane is perpendicular to their corresponding separating axis as Fig 3.

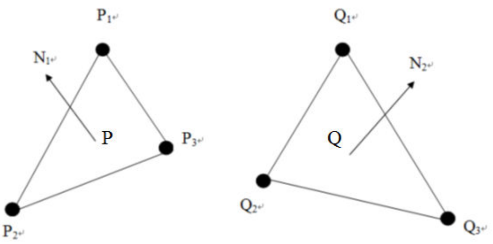

Fig. 3 Illustration for two triangular facets

Separating axes for coplanar case are follows,

$$\overline{N1} \times \overline{P1P2} \quad (1)$$

$$\overline{N1} \times \overline{P2P3} \quad (2)$$

$$\overline{N1} \times \overline{P3P1} \quad (3)$$

$$\overline{N2} \times \overline{Q1Q2} \quad (4)$$

$$\overline{N2} \times \overline{Q2Q3} \quad (5)$$

$$\overline{N2} \times \overline{Q3Q1} \quad (6)$$

Separating axes for non-coplanar case are $\overline{N1}$, $\overline{N2}$ and follows,

$$\overline{P1P2} \times \overline{Q1Q2} \quad (7)$$

$$\overline{P1P2} \times \overline{Q2Q3} \quad (8)$$

$$\overline{P1P2} \times \overline{Q3Q1} \quad (9)$$

$$\overline{P2P3} \times \overline{Q1Q2} \quad (10)$$

$$\overline{P2P3} \times \overline{Q2Q3} \quad (11)$$

$$\overline{P2P3} \times \overline{Q3Q1} \quad (12)$$

$$\overline{P3P1} \times \overline{Q1Q2} \quad (13)$$

$$\overline{P3P1} \times \overline{Q2Q3} \quad (14)$$

$$\overline{P3P1} \times \overline{Q3Q1} \quad (15)$$

B. Interpolating for Separating Axis

Each axial table for five-axis CNC machine is moving continuously during machining. In order to detect precisely, each CNC motion step, usually 0.01 sec/step, needs to perform 15 cross-product operations, (1)~(15), which are very heavy loaded in computation. This research uses the interpolating method that requires calculating the separating axes for the beginning and ending only. The separating axes for each motion step can be obtained by linear interpolating that reduces the computation time tremendously.

C. Design and Implementation of Web Learning Tool

The architecture of the proposed web based tool for collision detection is shown in Fig. 4. Since Web service is a powerful tool for many users, we use Apache as base to build our web based tool and then integrate it with Open Graphics Library (OpenGL) to simulate CNC collision detection, as shown in Fig. 5.

First, we build the three dimensions (3D) model of CNC machine with OpenGL following the specification of a real CNC machine. Second, our tool is scalable; that is, it can zoom in or out with mouse. Third, we implement a frond end with user interface (UI) to read simplest three dimension representation file format, STereoLithography (STL). STL is widely used for rapid prototyping to describe only the surface geometry of a three dimensional object without any representation of color, texture or other common CAD model attributes. Four, the motion of each axis is controlled by g-code and related parameters.

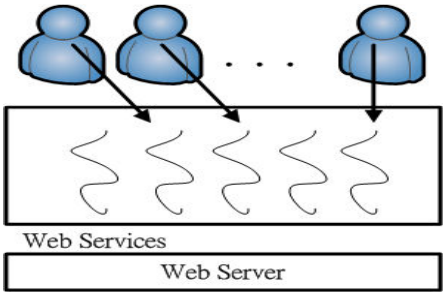

Fig. 4 Web learning tool

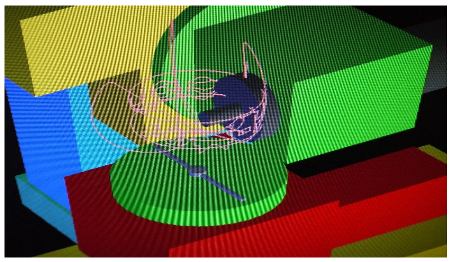

Fig. 5 3D OpenGL simulation

Notice that OpenGL is a cross-language, multi-platform application programming interface (API) for 3D graphics. On one hand, since OpenGL belongs to multimedia application, its performance can be improved by graphic processing unit (GPU). On the other hand, our web-based collision detection tool can also benefit from GPU because some network services can be parallelized by GPU. As a result, most mobile devices such as smart phones and PCs can use our tool on Windows and Linux. When users use our tool on Linux, it can be a basic GUI design for new CNC applications. For mobile devices, our tool is very easy to be compiled as an App. Even for cloud systems, it is also easy to become a cloud service for cloud clients.

Our tool can be easily to be downloaded by users from Internet. All related parameters have been set on it and users can see them easily from GUI. Users can set up all parameters or use some example to try it without cost. To pass data between different devices, we use a web page to set parameters and then translate data to eXtensible Markup Language (XML) format, XML-based information as shown in Fig. 6. This technique let users can perform their jobs everywhere. Besides, users can observe the result and monitor the execution status such as errors and warning messages as shown in Fig. 7.

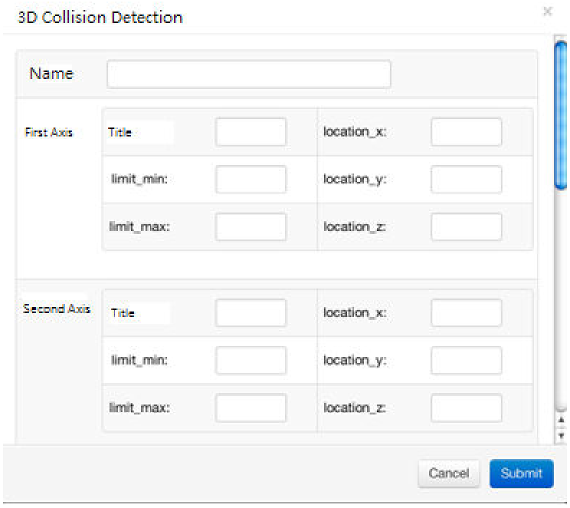

Fig. 6 Tool parameter

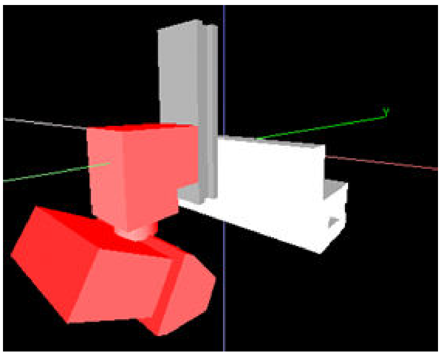

Fig. 7 3D Collision Detection

D. Computation Efficiency Analyses

This research uses the five-axis machine tool, as shown in Fig. 1, for collision detecting demonstration example. To implement the above collision detection algorithm, a simulation program was produced, using VB programming software. The computer used for this demonstration is 3.0 GHz AMD Athlon CPU. Fig. 8 shows the solid model of the demonstrating example, which colors the STL model for clear visualization.

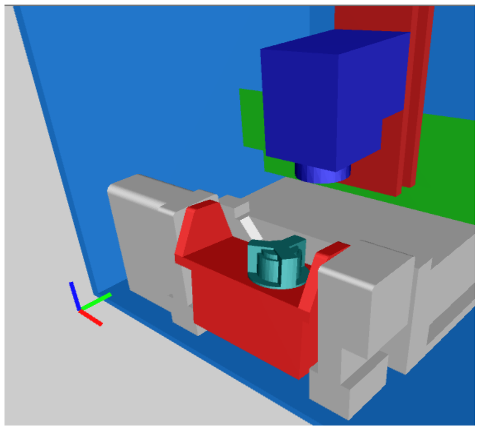

Fig. 8 The solid model for five-axis machine tool simulation

The testing g-code for five-axis motion is “G01 X300 Y-200 Z-200 A-3 Z-3 F15000”. The F15000 (mm/min) in the g-code means the feedrate that is the cutting tool moving velocity relative to the workpiece. The computation time for conventional method is shown in Fig. 9 that the detecting time for each five-axis motion is around 1800msec. The horizontal axis is the CNC sampling step that is 0.01sec/step. On the other hand, our approach, using interpolating method, takes 400 msec for each five-axis motion. Fig. 10 shows the recorded computation time for collision detections. The computation efficiency improves 4.5 times, comparing to the conventional method.

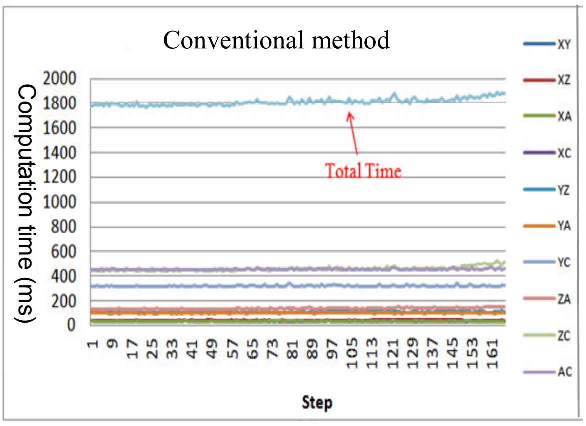

Fig. 9 The total computation time for each motion step, using conventional method

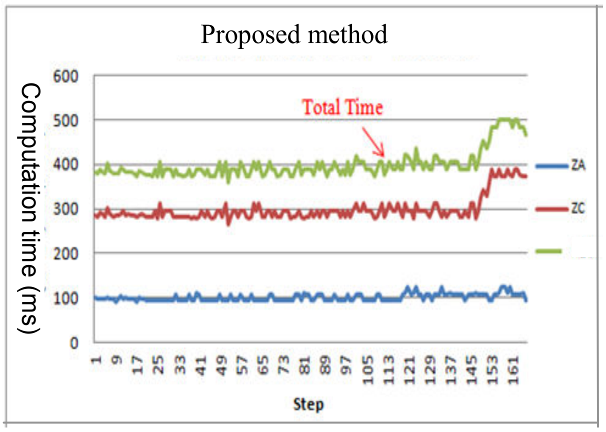

Fig. 10 Using proposed method

IV. Conclusions

This paper presents an efficient method for axial table collision detection on five-axis machining. Our approach includes 1. the analysis for machine kinematics that eliminates the unnecessary collision detection; 2. interpolating for separating axes that gets rid of complicated cross-product computation; and 3. efficient web-based collision detection tool to exchange experiences. Therefore, the demonstrated example shows that it improves 4.5 time of computation efficiency, comparing to the conventional detection method. Finally, we use GPU to improve performance of our tool so that it can execute collision detection smoothly to support cloud computing.

Acknowledgment

The authors would like to acknowledge the financial support of the National Science Council, Taiwan, R. O. C. under the grant, NSC 101-2221-E-194-021-MY3.

References

[1] I. Palmer and R. Grimsdale, "Collision detection for animation using sphere-trees," Computer Graphics Forum, Vol. 14, no. 2 (1995), pp. 105-116.
[2] G. Bradshaw and C. O'Sullivan, "Adaptive Medial-Axis Approximation For Sphere-Tree Construction," ACM Translations on Graphics, Vol. 23, no. 1 (2004), pp. 1-26.
[3] J. D. Cohen, M. C. Lin, D. Manocha, and M. Ponamgi, "I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scale Environments," ACM Interactive 3D Graphics Conference (1995), pp. 189-196.
[4] V. Bergen, "Efficient Collision Detection of Complex Deformable Models Using AABB Trees," Journal of Graphics Tools, Vol. 2, no. 4 (1997), pp.1-14.
[5] S. Gottschalk, M. Lin, and D. Manocha, "OBB-tree: A Hierarchical structure for Rapid Interference Detection," In: Proc. SIGGRAPH (1996), pp. 171-180.
[6] J. Chang, W. Wang, and M. Kima, "Efficient Collision Detection Using a Dual OBB-Sphere Bounding Volume Hierarchy," Computer-Aided Design Vol. 42 (2010), pp. 50-57.
[7] R.D. Owston, "The World Wide Web: A Technology to Enhance Teaching and Learning?", In American Educational Research Association, 1997.
[8] M.F. Shiratuddin, W. Thabet, "Virtual office walkthrough using a 3D game engine", International Journal of Design Computing, Volume 4, 2002.
[9] M.Woo, J. Neider, T. Davis, D. Shreiner, "OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2", In 3rd Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA 1999.
[10] Knaus WA, Draper EA, Wagner DP, Zimmerman JE. APACHE II: a severity of disease classification system. Crit Care Med 1985;13:818-829.
[11] W.T. Sung, S.C. Ou, Web-based learning in the computer-aided design curriculum, In Journal of Computer Assisted Learning, 18 (2) (2002), pp. 175–187.