Title:
Simulation of a Robot Machining System Based on Heterogeneous-Resolution Representation

Authors:
Yonghua Chen, yhchen@hku.hk, The University of Hong Kong
Ying Wei, wei.y1989@gmail.com, The University of Hong Kong

Keywords:
Robot machining, path planning, collision detection, simulation, heterogeneous-representation

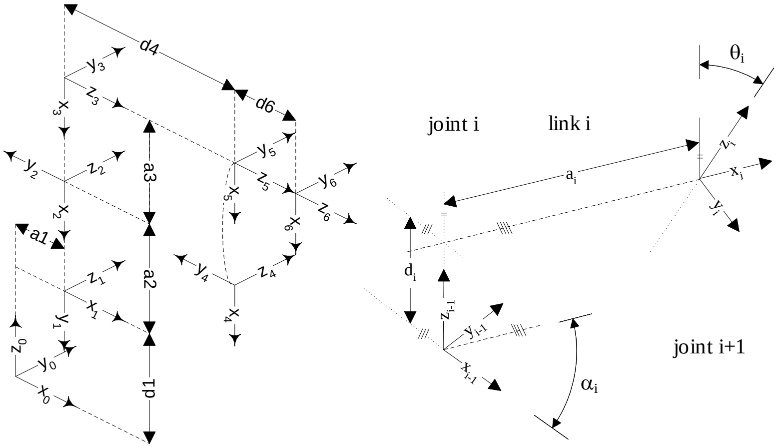
DOI: 10.14733/cadconfP.2015.416-420

Introduction:
Collision avoidance is a frequently encountered problem in machining processes, especially in robot-based machining. In a robot machining system, collision may occur between the robot arm, the tool, tool holder and the work piece together with its fixtures. Therefore, a precise collision detection algorithm is critical to ensure that the tool path is collision free, particularly in the area where the tool has contact with the work piece. To verify tool paths, a simulation system is developed. In the proposed system, the robotic system is modeled as a Constructive Solid Geometry (CSG) tree where the Denavit-Hartenberg (D-H) notation is applied to represent the robot arm transformation matrix. A proposed heterogeneous-resolution method is used to represent the work piece in which the area of the work piece near the current machining contact point is represented by triangular facets at a controlled accuracy whereas other parts of the work piece are described by Grid Height Array (GHA). This will save computation time in simulation. The proposed method is implemented to test the feasibility and effectiveness of the algorithm.

In modern industries, robotic arms have been widely used not only in their traditional applications areas such as pick and place, welding, etc., but also used in large part machining and grinding [5]. In any robotic applications, path planning and simulation is very important as it determines if an expected task can be done satisfactorily and safely or not [12]. A major part of robotic simulation is the detection and avoidance of collision. The most popular method of them is by using B-Rep to describe the objects that need to be checked, and then verifying if the edges of an object piecing into another object at discrete time instance. Some improved methods use Spatial-Occupancy Enumeration (SOE) to address the problem of reducing the number of pairs of objects or primitives that need to be checked. Octree [9], k-d tree [13], BSP-tree, Boxtree [3], OBBtrees [7], B-Rep indices [11], tetrahedral meshes and regular grids [6] are examples of SOE methods to solving this problem. Boundary Volumes is also a popular method in collision detection [8,10]. Sweep volume [1] is based on the idea that when an object moves along a given line or curve, the maximum space is the swept volume in a given time interval. Adding time as a dimension, Cameron [4] developed a four-dimension collision detection method based on sweep volume. Based on the four-dimension method, a vertex algorithm was developed by Aliyu [2].

System Overview:
The collision detection developed in this paper is applied to a robot machining system for large scale object rapid prototyping. The robotic system consists of an ABB IRB1400 articulated robot with six-degree-of-freedom mounted on a two-meter long linear track. With this configuration, the robot can cover a working envelope of 4M (Length) x 2M (Width) x 2M (Height). Several fixtures are installed in

the working platform for holding work-pieces, such as, a clamp, a rotary table, etc. In the current system, collision might occur among the robot arm, the tool holder, and the work piece. Accurate collision detection is required especially for the tool and the work piece since they are very close in the machining process.



(a) Kinematic configuration of IRB 1400 robot

(b) D-H link coordinate system and its parameters

Fig. 1: Kinematic configuration of the robot.

To describe the robot arm kinematics, the Denavit-Hartenberg notation and its derivations are used as in Fig.1. The list of the D-H parameters of each joint of the robot and the tool is given in Table 1.

Joint No. i	α_i	a_i	d_i	θ_i	Min Angle	Max Angle
1	-90	150	475	0	-170	170
2	0	-600	0	0	-70	70
3	90	-120	0	0	-65	70
4	-90	0	720	0	-150	150
5	90	0	0	0	-115	115
6	0	0	85	0	-300	300
Tool	0	100	200	0	-	-

Tab. 1: D-H parameters of the robotic joints.

Heterogeneous-resolution representation:

In STL representation, all geometric features are represented by triangles. Determining a triangle has intersection with an object or not is very simple. But a STL file is usually consisted of tens (or hundreds) of thousands of triangles. If every triangle is checked in the calculation, it is very computation expensive. In the proposed method, a method called heterogeneous-resolution is applied to increase the efficiency of collision detection.

A Grid Height Array (GHA) model can be described at a given accuracy using simple cuboids. GHA can be used as the model representation in collision detection, but in the area around the tool contact point, the accuracy of GHA is not high enough for collision detection. To be more precise in collision detection, the original STL representation is used. Since the number of the triangles in a STL file is very large, it is desirable to remove triangles that are unlikely to cause any collision at a given tool position. Here the proposed heterogeneous-resolution representation is used for this purpose. In the proposed

Proceedings of CAD'15, London, UK, June 22-25, 2015, 416-420

heterogeneous-resolution representation, the surface of the model close to the tool and tool holders is represented by its original triangular facets from the STL file while the rest is represented by GHA which is a much simpler representation and require much less memory space.

In R^3, finding the relations between the triangles and the moving tool is very difficult. In the proposed method, the relationship of the tool and the triangular surfaces is found by projecting them onto the tessellated plane of GHA, then grids on the plane are taken as a media to identify them. The following description focuses on the three steps in finding the relevant triangles:

1. Project triangles of the whole STL file onto the tessellated plane of GHA and identify the occupied grids for each triangle;
2. Project the tool onto the same plane and find the intersected grids which should be represented by triangles;
3. For each grid relevant to the tool on the projection plane, find the relevant triangles.

With the GHA representation in **Equation (1)**,

$$G = h_{m*n} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ h_{m1} & \cdots & \cdots & h_{mn} \end{bmatrix} \quad (1)$$

where m is the number of grids in row and n is the number of grids in column, $m \times n$ is the number of grids in this GHA.
The grids set can be described as

$$G^F = \{G_{fg} \,|\, [V_{fg}^1, V_{fg}^2, V_{fg}^3, V_{fg}^4] : f = 1,2,\cdots m,\ g = 1,2,\cdots n\} \quad (2)$$

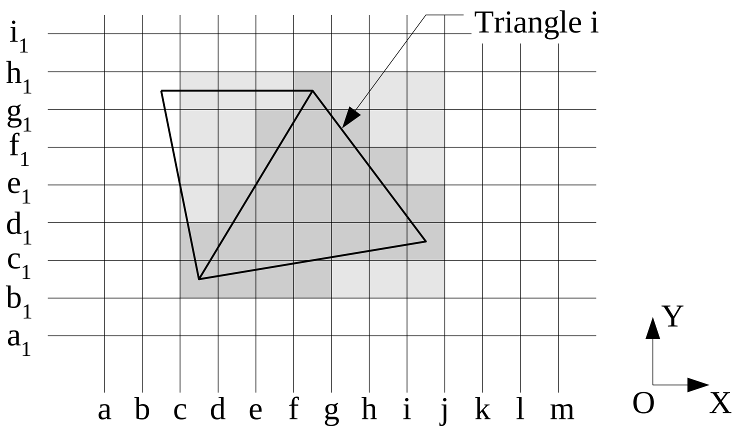


Fig. 2: Projecting STL file to the tessellated plane.

Suppose the machining orientation is along $\mathbf{Z}$ axis. **Fig.2** shows two projected triangles in the mesh and the projected triangle set is

$$S^p = \{Tr_i \,|\, Tr_i = [V_i^1, V_i^2, V_i^3, \mathbf{n}_i]^T, i = 1 \cdots m_t\} \quad (3)$$

where $V_i \in \mathcal{R}^2$, m_t is the number of triangles.

Proceedings of CAD'15, London, UK, June 22-25, 2015, 416-420

To find the relevant grids, the maximum and minimum $\mathbf{X}$ and $\mathbf{Y}$ value of the three vertices of the triangle are found. To triangle i, the gray area cjb_1h_1 in **Fig.3** represents the possible relevant grids as:

$$G_i^{RP} = \{^{rp}G_{pq}^i \mid [V_{pq}^1, V_{pq}^2, V_{pq}^3, V_{pq}^4]: p \le m,\ q \le n\} \tag{4}$$

All vertices of these possible relevant grids are checked to find if any of them are inside the triangle. Relevant grids are found by the following cases:

$$G_i^P = \{^pG_{pq}^i \mid [V_{pq}^1, V_{pq}^2, V_{pq}^3, V_{pq}^4]: p \le m,\ q \le n\} \tag{5a}$$

while $V_{pq}^1 \in Tr_i$ or $V_{pq}^2 \in Tr_i$ or $V_{pq}^3 \in Tr_i$ or $V_{pq}^4 \in Tr_i$;

$$G_i^P = G_i^{RP}\ ,\ p \le m\ ,\ q \le n \tag{5b}$$

where $V_{pq}^1 \notin Tr_i$ or $V_{pq}^2 \notin Tr_i$ or $V_{pq}^3 \notin Tr_i$ or $V_{pq}^4 \notin Tr_i$.

For instance, the relevant grids of triangle i are found as deep gray area in **Fig.2**.
Then, the tool is projected to the same plane. Usually, the fixtures of the tool are much higher than the model surface features, so only the tool, tool shank and spindle are considered here. Because the tool orientation is parallel to the $\mathbf{Z}$ axis of the coordinate, so the projection of the tool is a circle whose radius is the radius of the spindle as **Fig.3**. Using the same algorithm as above, the relevant grids are got as the deep gray area as

$$G^{TP} = \{G_{st}^{tp} \mid [V_{st}^1, V_{st}^2, V_{st}^3, V_{st}^4]: s \le m\ t \le n\} \tag{6}$$

With the grids selected by tool, the relevant triangles are selected. To each grid G_{st}^{tp} in G^{TP} and each triangle Tr_i in S^p,

$$K = G_{st}^{tp} \cap G_i^P, s < m, t < n \tag{7}$$

where G_i^P is the grids occupied by Tr_i. If $K \ne \Phi$, this triangle is marked and should appear in the multi-resolution model. By GHA and the identified triangles, the original model is represented at a heterogeneous-resolution accuracy as

$$M = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n} \\ h_{2,1} & h_{2,2} & \ddots & \vdots \\ \vdots & \ddots & T & \vdots \\ h_{m,1} & \cdots & \cdots & h_{m,n} \end{bmatrix} \tag{8}$$

In this model, the concerned area T, which is composed of grids in G^{TP}, is represented by the original STL file. The rest is described by GHA, which is a just a collection of cuboids.

Fig.4 shows an example model by heterogeneous-resolution. In Fig.4a, a simple STL model is shown where in Fig.4b, c, d, e, the model is described by heterogeneous-resolution according to the tool position. In Fig.4b, c, d, e, the area under the tool which is supposed to be the area under machining is described by triangles from STL files where the rest are represented by GHA. Thus, high accuracy (about 0.01mm) can be easily got in the area under the tool.

<u>Conclusions:</u>
The proposed collision detection method is based on the heterogeneous-resolution representation of a work piece model. Compared to tradition methods, it is a simple and precise solution. Given a machining task, simulations based on automatically generated G-code and robot configuration parameters can be performed before the actual machining takes place.

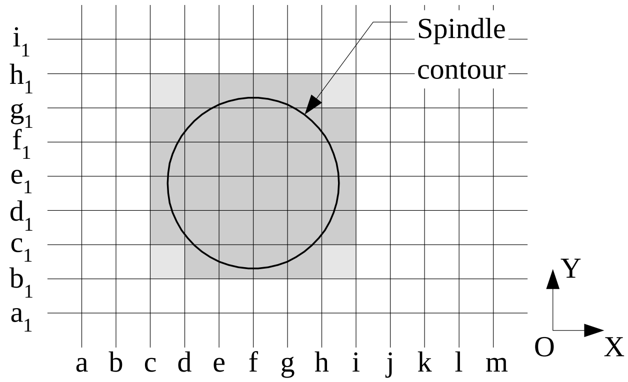


Fig. 3: Projecting tool on the tessellated plane.

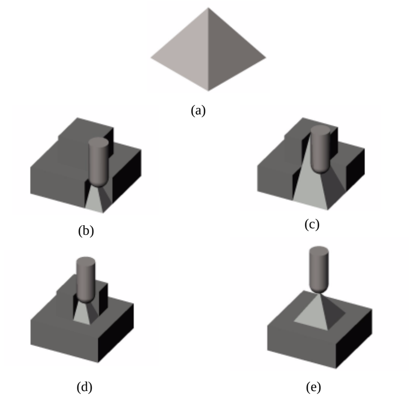


Fig. 4: Multi-resolution representation.

References:

[1] Abdel-Malek, K.; Yeh, H.-J.; Othman, S.: Sweep volumes: void and boundary identification, Computer Aided Design, 30(13), 1998, 1009-1018. http://dx.doi.org/10.1016/S0010-4485(98)00054-2

[2] Aliyu, M.-D.-S.; Al-Sultan K.-S.: Fast collision detection in four-dimensional space, European Journal of Operational Research, 114, 1999, 437-445. http://dx.doi.org/10.1016/S0377-2217(98)00030-7

[3] Barequet, G.; Chazelle, B.; Guibas, L.-J.; Mitchell, J.; Tal, A.: BOXTREE: a hierarchical representation for surfaces in 3D, Graphics Forum, 1996, 387–396. http://dx.doi.org/10.1111/1467-8659.1530387

[4] Cameron, S.: Collision detection by four-dimensional intersection testing, IEEE transactions on robotics and automation, 6(3), 1990, 291-302. http://dx.doi.org/10.1109/70.56661

[5] Chen, Y.-H.; Deng, F.-H.: Robot Machining: Recent Development and Future Research Issues, International Journal of Advanced Manufacturing Technology, 66(9-12), 2012, 1489-1497. http://dx.doi.org/10.1007/s00170-012-4433-4

[6] Garcia, A.-A; Serrano, N; Flaquer, J.: Solving the collision detection problem, IEEE Computer Graphics Application, 1(4), 1994, 36–43. http://dx.doi.org/10.1109/38.279041

[7] Gottschalk, S.; Lin, M.-C.; Manocha, D.: OBBtree: A hierarchical structure for rapid interference detection, Proceedings of SIGGRAPH'96, 1996. http://dx.doi.org/10.1145/237170.237244

[8] Hubbard, M.: Collision detection for interactive graphics applications, IEEE Transaction of Visual Computer Graphics, 1(3), 1995, 218-230. http://dx.doi.org/10.1109/2945.466717

[9] Noborio, H.; Fukuda, S.; Arimoto, S.: Fast interference check method using octree representation, Advanced Robotics, 3(3), 1989, 193–212. http://dx.doi.org/10.1163/156855389X00091

[10] Palmer, I.-J; Grimsdale, R.-L.: Collision detection for animation using sphere-trees, Computer Graphics Forum, 14(2), 1995, 105–116. http://dx.doi.org/10.1111/1467-8659.1420105

[11] Vanecek, G. Jr.: Brep-index: a multidimensional space partitioning tree, International Journal of Computer Geometry Applications, 1(3), 1991, 243–26. http://dx.doi.org/10.1142/S0218195991000189

[12] Vosniakos, G.; Matsas, E.: Improving feasibility of robotic milling through robot placement optimization, Robotics and Computer-Integrated Manufacturing, (26), 2010, 517-525. http://dx.doi.org/10.1016/j.rcim.2010.04.001

[13] Youn, J.-H.; Kohn, K.: Realtime collision detection for virtual reality applications, IEEE First Annual Virtual Reality Symposium, 1993. http://dx.doi.org/10.1109/VRAIS.1993.380750

Proceedings of CAD'15, London, UK, June 22-25, 2015, 416-420