Efficient concurrent toolpath planning for multi-material layered manufacturing

S.H. Choi and W.K. Zhu
Department of Industrial and Manufacturing Systems Engineering,
The University of Hong Kong, Hong Kong

Reviewed, accepted September 10, 2008

Abstract

This paper proposes an algorithm for planning efficient concurrent toolpaths to reduce the build-time of fabricating multi-material prototypes by layered manufacturing. The algorithm first sorts and partitions slice contours into hierarchical families of specific materials to enhance concurrent tool movements. It then detects overlapping of parametric tool envelopes with a "Two-phase Overlap Query Algorithm" to avoid potential tool collisions. Finally, it plans concurrent movements with an "Immediate Fabrication Algorithm" (IFA) to enhance fabrication efficiency by reducing idle time of tools. The algorithm is being implemented in a multi-material virtual prototyping system. It can be adapted for control of physical fabrication of multi-material prototypes when appropriate hardware becomes available.

1 Introduction

Today's market is characterized by intensifying global competition, increasing product complexity, continuous innovation dynamism, and harsh customer requirements. In order to survive and to maintain competitiveness, it is imperative for companies to strive to reduce the time-to-market and costs of product development.

A significant amount of development time and costs are often spent on product design and process validation. For this purpose, prototypes are commonly used for verification and testing of aesthetics, processes, and functionalities. Fabrication of a prototype is however an expensive and time-consuming undertaking.

Prototypes can be hard or soft. A hard prototype is a model which can be physically touched and manipulated, while a soft one is digitally simulated and may be regarded as a computer graphical image of the product.

A hard prototype is traditionally crafted by a skilled pattern-maker. It is slow and dimensionally inconsistent. About two decades ago, the stereolithography apparatus (SLA), generally known as the first commercial layered manufacturing (LM) or rapid prototyping (RP) machine, was developed. It is a relatively faster process that fabricates a physical prototype additively from a 3D CAD model layer by layer. LM systems are now widely used in product development and medical applications to save time and costs (Yildirim et al., 2006).

On the other hand, virtual prototyping (VP) is a process of using a digital prototype, in lieu of a physical one, for testing and evaluation of specific characteristics of a product or a manufacturing process. Virtual prototypes can be sent via the Internet to customers to solicit comments, or the process parameters can be tuned for optimal fabrication of physical ones. Thus

VP reduces the number of physical iterations and thereby the associated manufacturing overheads, leading to faster and cost-effective product development.

Currently, most commercial LM machines can only produce single-material prototypes. However, there is an increasing need for multi-material prototypes (Malone and Lipson 2007). It would be very desirable and useful to develop multi-material layered manufacturing (MMLM) technology. The material-depositing mechanism may consist of an array of nozzles or tools, each of which deposits a type of material on specific areas of a layer. Such a mechanism could be adapted from available hardware, but its control would require a relatively intelligent software system to accomplish effective and efficient fabrication of multi-material prototypes. Therefore, the development of MMLM technology is largely a software issue.

Indeed, processing of complex multi-material slice contours and planning of multi-toolpaths are particularly important issues of MMLM. Toolpath planning of MMLM may be sequential or concurrent. Concurrent toolpath planning saves the build-time of prototypes and hence shortens product development process. However, concurrent movements of tools may cause collision problems. It is desirable to develop an efficient concurrent toolpath planning algorithm for MMLM. This algorithm should allow as many concurrent tool movements as practicable without tool collision. It would be a significant step in development of MMLM for advanced product development.

2 Related works and research gap

Toolpath planning involves mainly two issues, namely tool collision detection and tool motion planning. Collision detection, also known as interference detection or contact determination, is a multi-discipline issue which is particularly important in computer graphics, CAD/CAM, motion planning, etc. (Ericson, 2005). Accuracy of detection and speed of computation are always two contradicting factors. The computational bottleneck of collision detection attracts many researchers. Many algorithms have been proposed and each of them has its merit and deficiency (Lin and Gottschalk, 1998).

Many researchers, particularly in areas of robotics and manufacturing, have contributed a lot to collision detection/avoidance and motion planning. Fraile et al. (2006) developed three methods to facilitate on-line path planning for multi-manipulator systems in dynamic environments with collision avoidance, where multi-manipulators may move concurrently in their common workspace. Liu and Kulatunga (2007) developed a simultaneous task allocation and motion coordination approach, which can solve the scheduling, planning and collision avoidance problems simultaneously. Kok et al. (2005) applied coordination graphs to a robotic domain by assigning tasks to the robots and then coordinating the different tasks. Ding et al. (2004) developed a global interference detection method for 5-axis machining of free-form surfaces. In order to simplify the process of updating tool positions and orientations in 5-axis machining, this method models a cutter and its holder by a hierarchy-oriented bounding box structure, whereas the workpiece surfaces are approximated by an octree.

In the domain of layered manufacturing, some research outputs of toolpath planning have been published. Zhu and Yu (2002) proposed a toolpath generation method of multi-material assembly for rapid manufacturing. With this method, a dexel-based "spatio-temporal" modelling approach was proposed for collision detection and concurrent tool movement. Lee and Kim (2006) proposed a multi-robot cooperation-based mobile printer system. A user uses this system to draw a picture on an input window of the GUI, and then the host computer commands client printer-robots to reproduce the same on a paper in a finite time. Malone and Lipson (2007) developed a set of software and hardware, called Fab@Home, which can sequentially fabricate multi-material prototypes of specific materials.

However, most of these toolpath planning algorithms are either sequential or can only handle relatively simple multi-material prototypes. Choi and Cheung (2005) developed a multi-material virtual prototyping system and a topological hierarchy-based approach to toolpath planning for MMLM. They later enhanced the toolpath planning algorithm by a parametric tool-envelope method (2006). Based on sorting and grouping toolpath-sets of specific materials, this method can generate concurrent toolpaths for relatively more complex objects. However, there may be considerable idle time of tools and the algorithm assumes constant tool speed. This paper presents the work in attempt to further enhance this concurrent toolpath planning algorithm. It is aimed that the algorithm could schedule for as many concurrent nozzle movements as practicable, and could handle variable tool speed. The algorithm is expected to substantially shorten the build-time of MMLM and hence the product development process accordingly. Moreover, the algorithm could be employed in other fields, like multi-robot motion planning and multi-arm robot assembly process.

3 Demonstration of proposed algorithm

3.1 Previous works

This section first demonstrates how the topological hierarchy-sorting approach to toolpath planning for MMLM works, and then presents the enhancements.

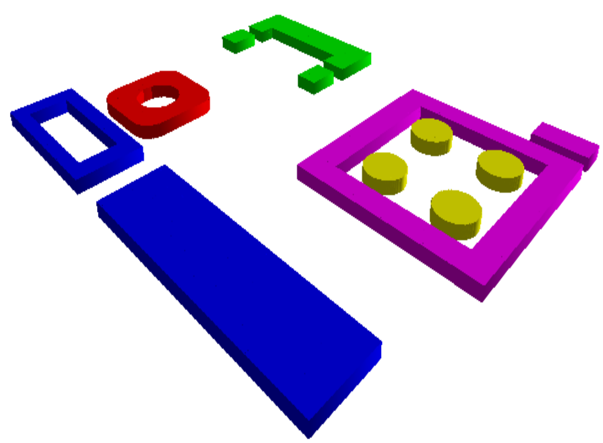


Figure 1 A discrete multi-material machine part

Figure 1 is a CAD model of a discrete multi-material machine part. Using the slicing and sorting modules of our in-house system (Choi and Kwok, 2002), we first slice the part into some layers and sort the topological hierarchy relationship of slice contours. A slice of the model is shown in Figure 2. Contours are categorized into three levels (level 0 to level 2), from the outermost contours to the innermost ones. For example, in envelope E_7, which encloses one or more families of contours, Contour C_9 (at level 0) is parent of C_{10} (at level 1) and they form a "contour family (pink)". C_{12} forms a contour family (yellow) by its own and so do C_{13}, C_{14}, and C_{15}, and they are enclosed in E_9, E_{10}, E_{11}, and E_{12} (all in yellow), respectively.

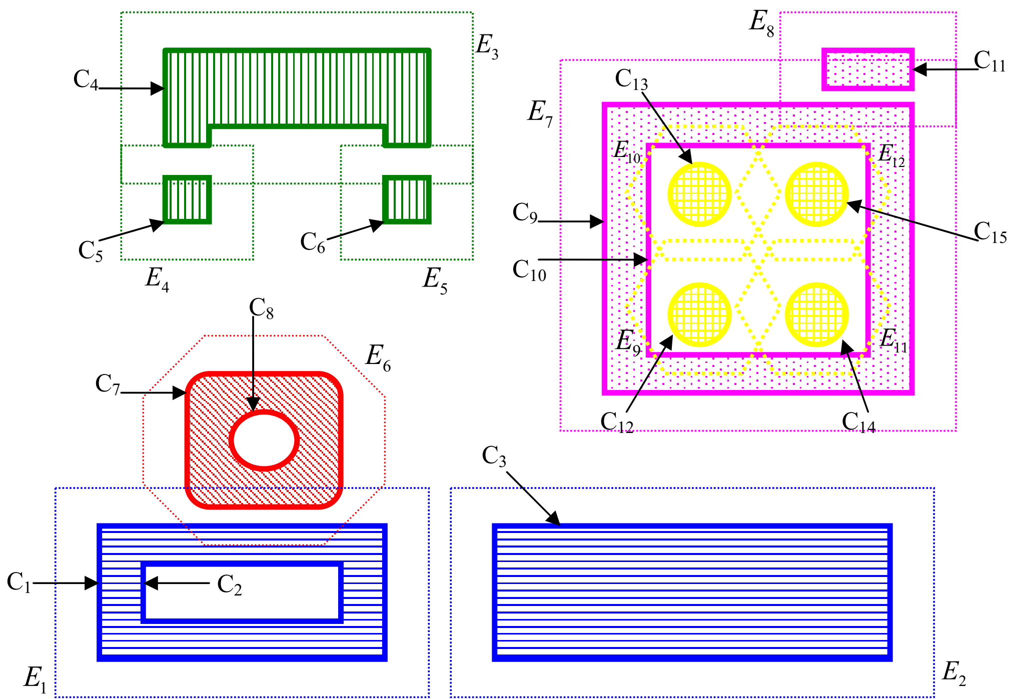


Figure 2 A slice of the machine part in Figure 1 (E_i for envelope and C_i for contour)

Topological hierarchy relationship of contours			Material
Parent-and-child list for contour containment	Contour families	Toolpaths	
Level 0: C_1 (E_1); C_3 (E_2); C_4 (E_3), C_5 (E_4), C_6 (E_5); C_7 (E_6); C_9; C_{11} (E_8)	1. $C_1 \rightarrow C_2$	$P_{C1,2}$	Blue
Level 1: C_2 (E_1); C_8 (E_6); C_{10}	2. C_3	P_{C3}	Blue
Level 2: C_{12} C_{13} C_{14} C_{15} (E_7)	3. C_4	P_{C4}	Green
	4. C_5	P_{C5}	Green
	5. C_6	P_{C6}	Green
	6. $C_7 \rightarrow C_8$	$P_{C7,8}$	Red
	7. $C_9 \rightarrow C_{10}$	$P_{C9,10}$	Pink
Group related toolpaths as toolpath-sets for association with a related nozzle based on material property:	8. C_{11}	P_{C11}	Pink
Set 1 ($P_{C1,2}+P_{C3}$) → N_1 (nozzle 1 for blue material)	9. C_{12}	P_{C12}	Yellow
Set 2 ($P_{C4}+P_{C5}+P_{C6}$) → N_2 (nozzle 2 for green material)	10. C_{13}	P_{C13}	Yellow
Set 3 ($P_{C7,8}$) → N_3 (nozzle 3 for red material)	11. C_{14}	P_{C14}	Yellow
Set 4 ($P_{C9,10}+P_{C11}$) → N_4 (nozzle 4 for pink material)	12. C_{15}	P_{C15}	Yellow
Set 5 ($P_{C12}+P_{C13}+P_{C14}+P_{C15}$) → N_5 (nozzle 5 for yellow material)			

Table 1 The topological hierarchy and toolpath-sets of the contours in Figure 2

A contour family of a specific material forms a toolpath. Inside E_7, there are six toolpaths, namely $P_{C9,10}$, P_{C11} (both in pink) and P_{C12}, P_{C13}, P_{C14}, P_{C15} (all in yellow). Toolpaths for a specific material are grouped together to form a toolpath-set. For example, toolpaths $P_{C9,10}$, and P_{C11} (both in pink) are grouped together to form toolpath-set 4 associated with nozzle N_4. Table 1 shows complete grouping of toolpaths. The outermost contour of each contour family is offset by a distance of the tool radius to form tool envelope. A tool envelope, in which only a tool is allowed to move along the related toolpath-set to deposit the specific material, is constructed for detection of potential tool collisions. If a pair of tool envelopes overlaps, there may be tool collisions in the envelopes. To balance the accuracy and speed of overlap query, the shape of a tool envelope is parametric, in that it can be varied from being a quadrangle that results in the least accurate but the fastest overlap query, to being an exact offset of the contour that entails the most accurate but the slowest overlap query. Table 2 shows the result of overlap queries. Based on this result, materials can be deposited concurrently by arranging the toolpath-sets into two groups, namely group 1 (blue, green, and pink) and group 2 (red and yellow), with the related nozzles associated accordingly as shown in Figure 3.

	Green (Set 2)	Red (Set 3)	Pink (Set 4)	Yellow (Set 5)
Blue (Set 1)	Non-overlap	Overlap	Non-overlap	Non-overlap
Green (Set 2)		Non-overlap	Non-overlap	Non-overlap
Red (Set 3)			Non-overlap	Non-overlap
Pink (Set 4)				Overlap

Table 2 Result of overlap queries of the previous algorithm

Step 1: nozzles N_1, N_2, and N_4 deposit the specified materials concurrently and then return to the datum point.

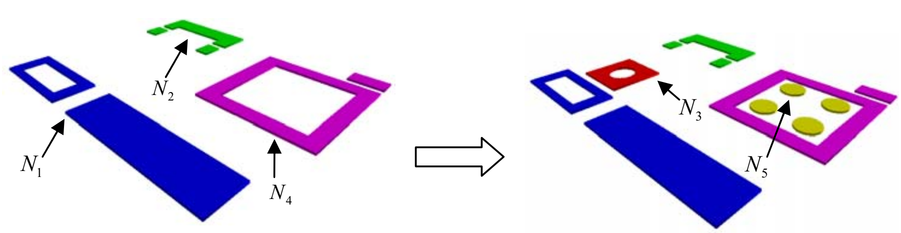
- N_1 (blue) for Set 1 ($P_{C1,2}+P_{C3}$)
- N_2 (green) for Set 2 ($P_{C4}+P_{C5}+P_{C6}$)
- N_4 (pink) for Set 4 ($P_{C9,10}+P_{C11}$)

⇒

Step 2: nozzles N_3, and N_5 deposit the specified materials concurrently and then return to the datum point.

- N_3 (red) for Set 3 ($P_{C7,8}$)
- N_5 (yellow) for Set 5 ($P_{C12}+P_{C13}+P_{C14}+P_{C15}$)

Figure 3 Concurrent nozzle movements by the previous algorithm



t<=8, nozzles N_1, N_2, N_4 deposit blue, green, and pink materials concurrently

8<t<=12, nozzles N_3, N_5 deposit red, and yellow materials concurrently

Figure 4 Virtual fabrication of the machine part by the previous algorithm

Figure 4 shows the virtual fabrication of the machine part by the previous algorithm. Unless specified otherwise, "t" stands for time and is in "time unit". Figure 5 shows the time chart of the previous algorithm. It can be seen that there is some idle time of tools. Take nozzle N_3 (red) and N_1 (blue) for example. In Figure 2, E_6 (red) overlaps with E_1 (blue). Since E_1 is grouped into toolpath-set 1 with E_2, nozzle N_3 (red) cannot start to deposit material until N_1 (blue) finishes depositing both E_1 and E_2. In other words, the time spent on depositing E_2 (blue) is the idle time of N_3 (red). In regard to nozzle N_5 (yellow), it should start depositing immediately after N_4 (pink) finishes depositing material in E_7 (pink). However, N_5 is grouped into toolpath-set 5 which is triggered in step 2, so it has to wait until the finish of toolpath-set 1 (step 1), that is, until N_1 (blue) finishes depositing E_2 (blue).

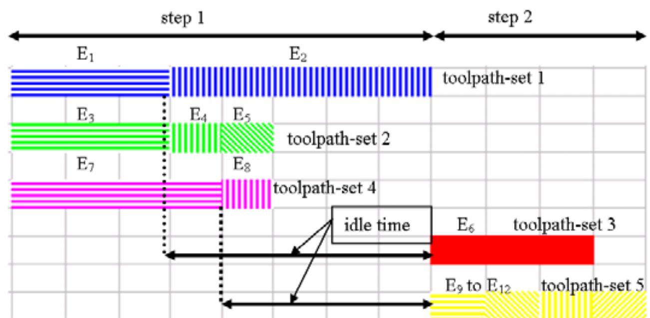


Figure 5 Time chart of previous algorithm

In order to reduce idle time of tools, we propose an enhanced concurrent toolpath planning algorithm called "Immediate Fabrication Algorithm" (IFA). This algorithm does not group toolpaths into toolpath-sets. Instead, it increases the number of tool envelopes (hence smaller envelopes) such that when a contour family (toolpath) in an envelope is finished, deposition of another contour family (toolpath) in the corresponding overlapping envelope can be more likely started immediately. However, due to increased number of envelope overlap queries, it is necessary to develop an efficient overlap query algorithm first, which is introduced in the following section.

3.2 Two-phase Overlap Query Algorithm (TOQA)

There are twelve envelopes in Figure 2. Pairwise overlap query of n envelopes requires $(n-1)+(n-2)+\ldots\ldots+2+1=n(n-1)/2=O[n(n-1)/2]$ tests in the worst case. Due to the quadratic time complexity, full enumeration of testing all envelope pairs becomes too expensive even for a moderate value of n. To speed up the process, the number of pairwise tests must be reduced. When a slice of a prototype is complex and the number of tool envelopes is big, time complexity of overlap query is lengthy, impairing the performance of real-time virtual prototyping simulation. To tackle this problem, a "Two-phase Overlap Query Algorithm" (TOQA), which consists of "material-group phase" and "envelope phase", is proposed as follows:

```
for layer Li      // The "Two-phase Overlap Query Algorithm"
{     "material-group phase" overlap query;
      if (overlap detected)
           "envelope-phase" overlap query;
}
```

When executing overlap query for a layer, the "material-group phase" is invoked first. It groups envelopes of the same material property together and then executes pairwise overlap query between material-groups. Time complexity for *m* materials is O[*m*(*m*-1)/2], where *m* is usually much smaller than *n*. If overlapping is detected during the "material-group phase", the "envelope phase" will be invoked to execute further pairwise tests between envelopes involved.

Now let's take Figure 2 for example and analyze the improved efficiency of the proposed algorithm. In Figure 2, there are *n*=12 tool envelopes for *m*=5 materials. When executing overlap queries of the layer, the "material-group phase" is invoked first. Pairwise tests between material-groups are executed and the time complexity is O[m(m-1)/2]=O[5*(5-1)/2]=O(10). The result of overlap queries of the material-group phase is shown in Table 3.

	Green	Red	Pink	Yellow
Blue	Non-overlap	Overlap	Non-overlap	Non-overlap
Green		Non-overlap	Non-overlap	Non-overlap
Red			Non-overlap	Non-overlap
Pink				Overlap

material-group phase

⇩

envelope phase

	E_1 (Blue)	E_2 (Blue)
E_6 (Red)	Overlap	Non-overlap

	E_7 (Pink)	E_8 (Pink)
E_9 (Yellow)	Overlap	Non-overlap
E_{10} (Yellow)	Overlap	Non-overlap
E_{11} (Yellow)	Overlap	Non-overlap
E_{12} (Yellow)	Overlap	Non-overlap

Table 3 Result of TOQA for Figure 2

The "material-group phase" overlap query algorithm finds that there are overlaps between material-groups blue and red, and between material-groups pink and yellow. So, further pairwise tests between envelopes involved are executed in the "envelope phase". For material-groups blue and red, the time complexity of pairwise tests for three envelopes is O[3*(3-1)/2]=O(3) while the test between E_1 and E_2 (of the same material) can be ignored. So here, the time complexity is reduced to O(3-1)=O(2) and overlapping of E_1 (blue) and E_6 (red) is found out as shown in Table 3 (envelope phase, left). For material-groups pink and yellow, there are two pink envelopes and four yellow envelopes. All-pairwise overlap query requires O[6*(6-1)/2]=O(15) tests. Ignoring a test between two pink envelopes (of the same material) and O[4*(4-1)/2]=O(6) tests between four yellow envelopes (of the same material), the time complexity is reduced to O(15-1-6)=O(8). It is found that E_7 (pink) overlaps with E_9, E_{10}, E_{11}, E_{12} (all are yellow), as shown in Table 3 (envelope phase, right). Together, the time complexity of the "envelope phase" for Figure 2 is O(2+8)=O(10).

Totally, the time complexity of the two-phase test is O(10+10)=O(20). Compared with that of full enumeration of testing all envelope pairs, O[n(n-1)/2]=O[12*(12-1)/2]=O(66), we achieve an improved efficiency of (66-20)/66=69.7%, as shown in Figure 6. Generally, since m is smaller than n, O[m(m-1)/2] is much smaller than O[n(n-1)/2]. Though further tests are required in "envelope phase", the complexity is significantly reduced. Meanwhile, ignoring tests between envelopes of the same materials can further reduce the time complexity. Indeed, the more contour families in a slice, the better efficiency of the proposed "Two-phase Overlap Query Algorithm". This is particularly important for virtual prototyping simulation.

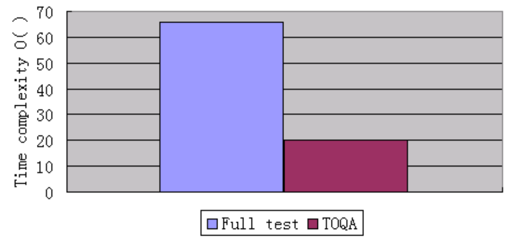


Figure 6 Comparison of time complexities of full test and TOQA

3.3 Immediate Fabrication Algorithm (IFA)

To reduce idle time of tools, we propose the "Immediate Fabrication Algorithm" (IFA). This algorithm does not group toolpaths into toolpath-sets. Instead, it improves concurrency of tool movements by enhancing the likelihood of triggering material deposition by the subsequent tool immediately after one finishes the contours in the overlapping envelope. The psuedocode of IFA is as follows:

```
IFA ( )      // Immediate Fabrication Algorithm (IFA)
{
   for layer Li {
         for envelope Ei {
               Two-phase Overlap Query; }
         group non-collision toolpaths;
         while (layer not finished) {
                  display concurrent toolpaths;
                  check flag of envelope status;
                     if (overlapped envelope finished) {
                           add a new nozzle to current toolpaths;
                           trigger the new nozzle;
                        }
         } //end of "while (layer not finished)"
    } // end of "for layer Li"
 } // end of "IFA ( )"
```

Take nozzle N_3 (red) and N_1 (blue) for example again. In TOQA, it is found that envelope E_6 (red) overlaps with E_1 (blue). So nozzle N_3 (red) stands by. When t=3 (Figure 7), N_1 (blue) has just finished depositing in E_1 (blue) and continues to E_2 (blue). Since E_6 (red) does not overlap with E_2 (blue), right at this moment (t=3), nozzle N_3 (red) is triggered to deposit material on contour family in envelope E_6 (red) immediately. In regard to nozzle N_5 (yellow), envelopes E_9, E_{10}, $E_{11,}$ and E_{12} (all in yellow) overlap with E_7 (pink) in TOQA, so nozzle N_5 stands by. When t=4 (Figure 8), N_4 (pink) has just finished depositing material in envelope E_7 (pink) and continues to deposit material on the contour family in E_8 (pink). Since E_9, E_{10}, $E_{11,}$ and E_{12} (all in yellow) do not overlap with E_8 (pink), right at this moment (t=4), nozzle N_5 (yellow) is triggered to deposit material on contour families in envelopes E_9, E_{10}, E_{11} and E_{12} (all in yellow) immediately.

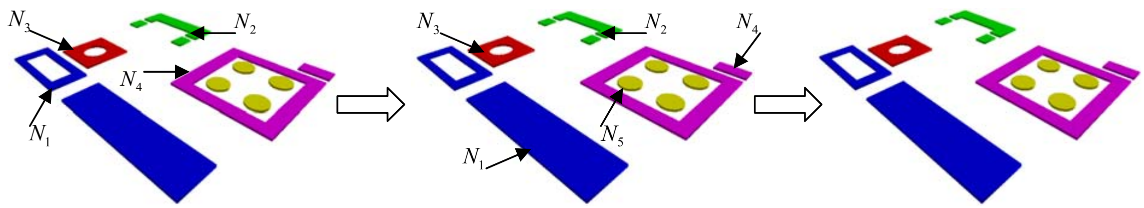


Figure 7 t=3, $N_{1,}$ $N_{2,}$ N_4 move concurrently. N_1 has just finished E_1 (blue) and continues to E_2 (blue); N_3 is triggered to start E_6 (red)

Figure 8 t=4, $N_{1,}$ $N_{2,}$ $N_{3,}$ N_4 move concurrently. N_4 has just finished E_7 (pink) and continues to E_8 (pink); N_5 is triggered to start E_9, E_{10}, E_{11} and E_{12} (all in yellow)

Figure 9 t=8, layer completed

Figures 7 to 9 show the complete fabrication process of IFA, while Figure 10 is the corresponding time chart. Comparing Figure 10 with Figure 5, it can be seen the build-time of the previous algorithm is twelve units while IFA gives eight, improving the efficiency by 33%, as shown in Figure 11.

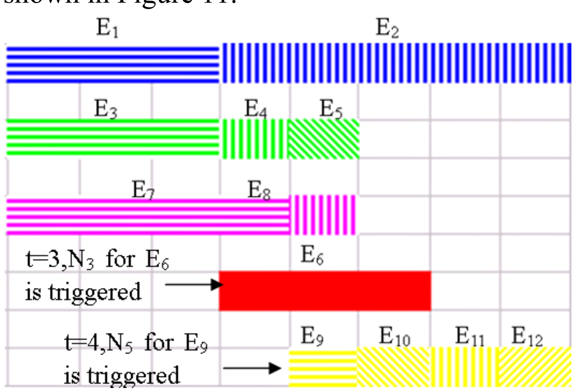


Figure 10 Time chart of IFA

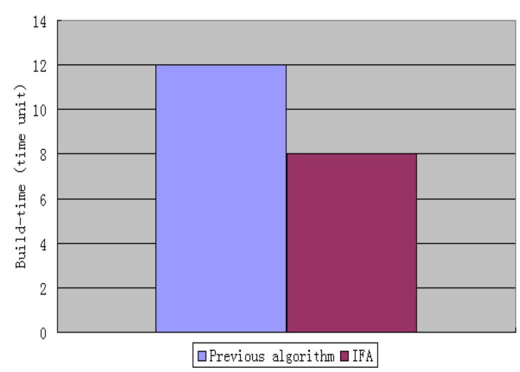


Figure 11 Comparison of build-times of the previous algorithm and IFA

4 Case study

Figure 12 shows a CAD model of a gearbox assembly and a slice of it. There are eight contours and four types of materials. The topological hierarchy relationship and the result of TOQA are shown in Tables 4 and 5, respectively. Comparing full enumeration test with the TOQA, time complexity of TOQA is only O(8) while full test takes O(21) as in Figure 13, improving the efficiency by 61.9%, which would be beneficial for real-time virtual prototyping simulation. Based on the result of TOQA, concurrent toolpaths are planned according to IFA.

First (t=0), nozzle N_1 (blue), N_2 (purple), and N_3 (brown) start to deposit the related materials in the specified envelopes concurrently (Figure 14). At t=3 (Figure 15), N_1 (blue) has not finished task while N_3 (brown) has just finished envelope E_5 (brown) and continues to E_6 (brown). At this moment, since E_7 (green) does not overlap with E_6, N_4 (green) is immediately triggered to start depositing material in E_7 without having to waiting for N_1 to finish. Figures 14 to 16 show the IFA process.

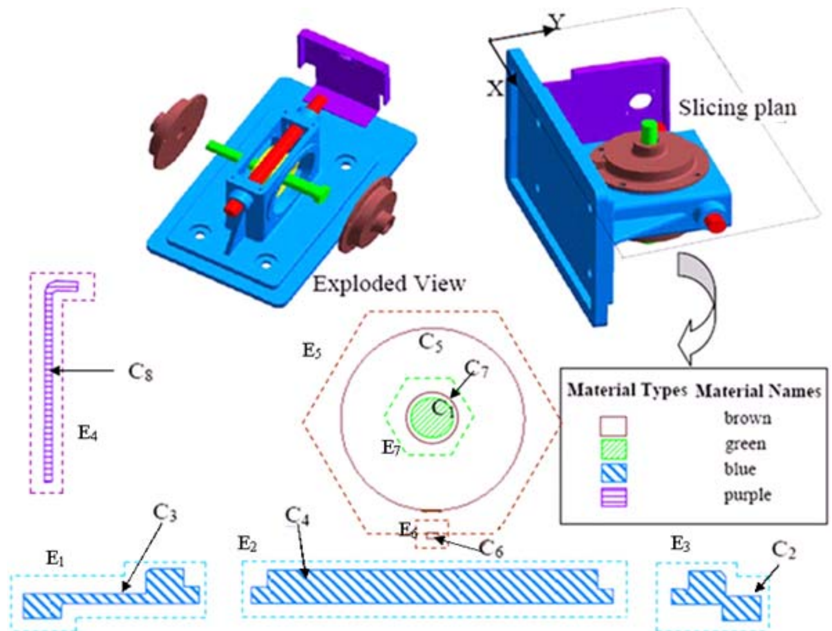


Figure 12 A CAD model a gearbox assembly and slice of it

Topological hierarchy relationship of contours										Material
Parent-and-child list for contour containment								Contour Families	Toolpaths	
Level 0	C_3 (E1)	C_8 (E4)	C_4 (E2)	C_5 (E5)	C_6 (E6)	C_2 (E3)		1. C_3	P_{C3}	Blue
								2. C_8	P_{C8}	Purple
Level 1				C_7				3. C_4	P_{C4}	Blue
								4. $C_5 \rightarrow C_7$	$P_{C5,7}$	Brown
Level 2				C_1				5. C_1	P_{C1}	Green
								6. C_6	P_{C6}	Brown
								7. C_2	P_{C2}	Blue

Table 4 Topological hierarchy relationship of contours in Figure 12

	Purple	Brown	Green
Blue	Non-overlap	Non-overlap	Non-overlap
Purple		Non-overlap	Non-overlap
Brown			Overlapverlap

material-group phase

⇨

	E_7 (Green)
E_5 (Brown)	Overlap
E_6 (Brown)	Non-overlap

envelope phase

Table 5 Result of Two-phase Overlap Query Algorithm for Figure 12

Figure 17 shows the time charts of the previous algorithm and IFA. N_4 (green) is triggered to deposit material in envelope E_7 (green) as soon as E_5 (brown) is finished, without having to wait for N_1 (blue) to finish. Here, the build-time of IFA is five units. In comparison with seven units of the previous algorithm, there is an improved efficiency of 28.6%, as shown in Figure 18.

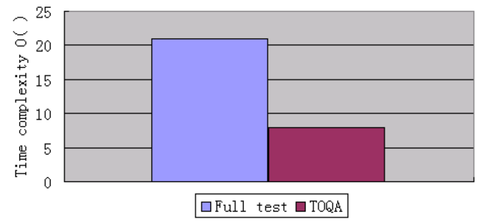


Figure 13 Comparison of time complexities of full test and "TOQA"

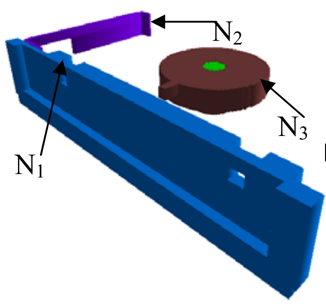


Figure 14 t=0, N_1, N_2, and N_3 start moving concurrently

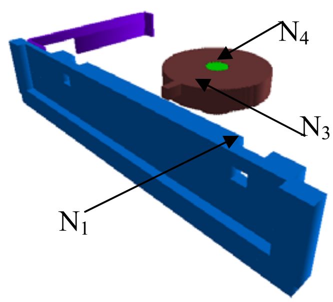


Figure 15 t=3, N_1 has not finished task; N_3 has just finished envelope E_5 (brown) and continues to E_6 (brown); N_4 is triggered to start E_7 (green)

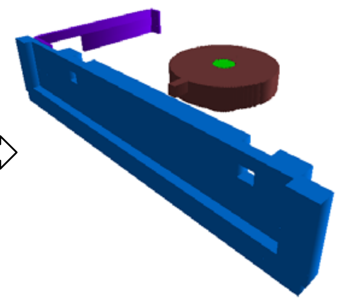


Figure 16 t=5, layer completed

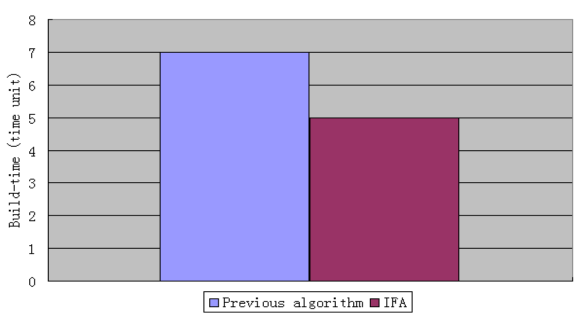


Figure 18 Comparison of the build-times of the previous algorithm and IFA

5 Conclusions and future works

This paper proposes an algorithm for improving the efficiency of concurrent toolpaths to reduce the build-time of MMLM prototypes. Based on the topological hierarchy of slice contours, it adopts the "Two-phase Overlap Query Algorithm" to speed up overlap query of parametric tool envelopes for detection of tool collisions. Subsequently, the "Immediate Fabrication Algorithm", is used to generate

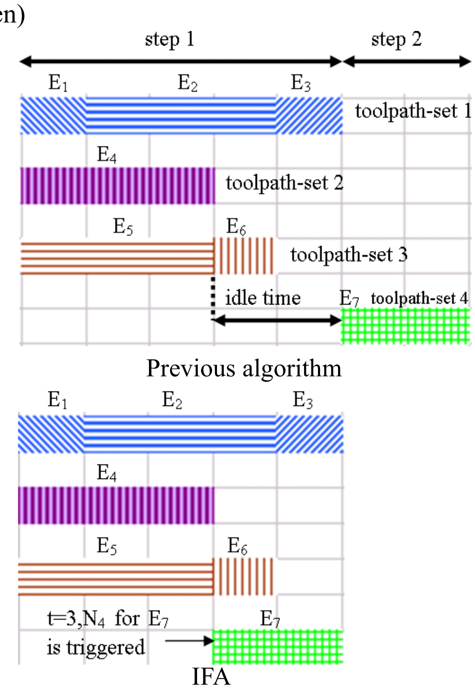


Figure 17 Time charts of the previous algorithm and IFA

efficient toolpaths to enhance concurrent movements by reducing idle time of tools. The algorithm is being implemented in a multi-material virtual prototyping system. It can be adapted for control of physical fabrication of multi-material prototypes when appropriate hardware becomes available. Currently, a tool envelope is constructed by offsetting the slice contour. If a pair of tool envelopes overlaps, tools in the tested envelopes may collide. This method is relatively a rough approximation. Future work would be dedicated to constructing envelopes around the tools, rather than around the slice contours. Collision detection between tool envelopes is executed at a certain time-interval while the tools are moving around. Direct motion planning of multi-tools would provide more accurate collision detection and improve the concurrency of tool movements. Moreover, in practical fabrication, it may be necessary to adjust the deposition speed for materials of different properties. These issues are generally concerned with "spatio-temporal" collaboration problems, which will be investigated in the near future.

6 References

Choi, S.H. and Cheung, H.H. 2005. A multi-material virtual prototyping system. *Computer-Aided Design,* 37(1):123-136.

Choi, S.H. and Cheung, H.H. 2006. A topological hierarchy-based approach to toolpath planning for multi-material layered manufacturing. *Computer-Aided Design,* 38(2):143-156.

Choi, S.H. and Kwok, K.T. 2002. Hierarchical slice contours for layered-manufacturing. *Computers In Industry,* 48(3):219-239.

Ding, S., Mannan, M.A. and Poo, A.N. 2004. Oriented bounding box and octree based global interference detection in 5-axis machining of free-form surfaces. *Computer-Aided Design,* 36(13):1281-1294.

Ericson, C. 2005. Real-time collision detection. Elsevier.

Fraile, J.C., Turiel, J.P. and Sanchez, J.L.G. 2006. Comparative analysis of collision-free path-planning methods for multi-manipulator systems. *Robotica,* 24(6):711-726.

Kok, J.R., Spaan, M.T.J. and Vlassis, N. 2005. Non-communicative multi-robot coordination in dynamic evironments. *Robotics and Autonomous Systems,* 50(2):99-114.

Lee, K.H. and Kim, J.H. 2006. Multi-robot cooperation-based mobile printer system. *Robotics and Autonomous Systems,* 54(4):193-204.

Lin, M.C. and Gottschalk, S. 1998. Collision detection between geometric models: a survey. In *Proceedings of IMA Conference on Mathematics of Surfaces*, volume 1, pp.602–608.

Liu, D.K. and Kulatunga, A.K. 2007. Simultaneous planning and scheduling for multi-autonomous vehicles. *Studies in computational intelligence,* 49(4):437-464.

Malone, E. and Lipson, H. 2007. Fab@Home: the personal desktop fabricator kit. *Rapid Prototyping Journal,* 13(4):245-255.

Yildirim, E. Yin, X., Guceri, S. and Sun, W. 2006. A Preliminary Study on Using Multi-Nozzle Polymer Deposition System to Fabricate Composite Alginate/Carbon Nanotube Tissue Scaffolds. In *Proceedings of 17th Solid Freeform Fabrication Symposium,* pp.84-95.

Zhu, W.M. and Yu, K.M. 2002. Tool path generation of multi-material assembly for rapid manufacture. *Rapid Prototyping Journal,* 8(5):277-283.