Machining large complex shapes using a 7 DoF device

J.S.M. Vergeest, J.W.H. Tangelder, Zs. Kovács, Gy. Kuczogi, and I. Horváth,
Faculty of Design, Engineering and Production Delft University of Technology

Keywords: Robot path planning, approach directions, collision avoidance, rapid prototyping

Abstract A new approach to the refixturing-free sculpturing of large, arbitrarily complex shapes has been developed. Based on an enhanced accessibility analysis of the target shape, a reduced set of candidate machining orientations is determined. Then, machining takes place at very close distance between the tooling machinery and the part, thus achieving effective material removal in complex regions. Minkowski operations on the involved (dynamic) volumes are performed to update their multi-map representations. Performance tests, both virtual and physical, show that it is possible to automatically generate a strategy for material removal and for tool path planning. The method has been implemented on in a 7 DoF workcell called the Sculpturing Robot, at the Faculty of Design, Engineering and Production of Delft University of Technology. Presently the system is applied to perform Rapid Prototyping of conceptual shape models.

1. INTRODUCTION

The major issues of machining complex shapes can be summarized as

1) Obtaining a smooth surface
2) Reduction of the machining time and of changing fixtures
3) Avoidance of any collision
4) Reduction of the computation
5) Automation of the machining planning

Depending on the practical application of machining, some of these requirements may become more or less important. For example, in die manufacturing the issues 2 and 5 are less relevant than are the issues 1 and 3. In order to ensure collision abse1nce, it may be decided to rely on extensive manual intersection testing using some graphical interactive technique. In contrast, high speed machining for rapid shape prototyping puts emphasis on issues 2 and 5.

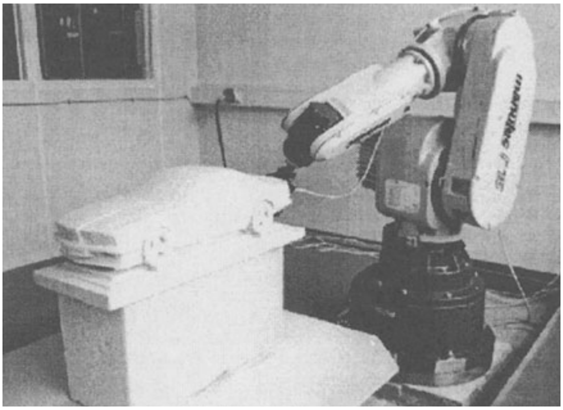

Figure 1. The Sculpturing Robot ia 7 DoF machining device. The 7th variable is the rotation angle of the tuntable, carrying the workpiece

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35392-0_40

G. J. Olling et al. (eds.), *Machining Impossible Shapes*

There is at least one application, namely finish-free shape prototyping, that involves the compliance with all of the five requirements simultaneously. The paper is dealing with this application, *i.e.*, the materialization of CAD models safely, rapidly, accurately and fully automatically, using machining technology. To produce accurate, finish-free models we need to reduce the scallops between adjacent tool paths (Choi, Park & Jun 1993), and hence, in general, have to generate dense tool paths. To make the fabrication process rapid and automatic we need to eliminate any refixturing, and hence have to introduce redundancy in the machining mechanism, *i.e.* supply additional degrees of freedom (DoF). Although collision avoidance can be achieved for 5-axis machining of sculptured surfaces (You & Chu 1997), this task becomes significantly harder if the number of DoF increases. From these considerations it may be obvious that the problem of rapid prototyping based on machining is far from being solved. The relevance to solve this problem is high since it is expected that for the fabrication of large-sized objects, machining technology could be superior compared to common RP methods such as material deposition or layer building (Tangelder 1998, Vergeest & Tangelder 1996, Wall, Ulrich & Flowers 1992). In this paper we present an approach to the problem of RP using machining, and a hardware and software implementation for practical use (figure 1).

2. DIVISION OF THE MACHINING PROBLEM INTO SUBPROBLEMS

The problem of RP using machining involves the specification of the process to remove material from an initial stock such that the remainder of the stock is a materialization of the design model's shape. In the following we will define this process in a basic form, although we made a number of provisions in the software to accommodate for inaccuracies and ambiguities in the input for the process. The design model F may specify any free-form solid object, or equivalently its boundary. The stock S is assumed to be, initially, a block that encloses F. Material is removed from S by a moving milling tool T. T is convex and a strict set-theoretical behavior is assumed for the process, *i.e.*, $S' = S - V$, where S' is the material left from S after T has swept the volume V. During the machining process T should obviously never intersect F. Furthermore, there are two types of collisions that should be detected and avoided: 1) collision of the machinery (e.g. a robot) R with the stock-in-progress S, and 2) collision of either R or T with any obstacle O. The former type of collision avoidance deserves special treatment since S is involved, which is a dynamic object; the free space increases over time as S erodes. Recently, a direct analysis of the erosion of free configuration space was made in order to support path finding for the machining process (Vergeest, Broek & Tangelder 1995). They achieved a partitioning of an (up to 7D) configuration space into a) free regions, b) later-to-be-free regions and c) forbidden regions. Projection of these regions onto 3D subspaces could provide hints for collision free path planning. However, it seems not feasible to apply this methodology to the practical problem of sculpturing. On the contrary, it is known that an optimal solution even to the so-called peg and hole problem is already hard to achieve directly in 6D space (Joskowicz & Taylor 1996).

In (Tangelder, Vergeest & Overmars 1998) it was proposed to subdivide the machining problem into an *accessibility problem* and a *path planning problem*. The former problem deals with the selection of (as few as possible) approach orientations from which it should be possible to reach the volume F. The path planning problem then deals with one of those orientations at a time, and performs high-precision material removal using a "just no collision" strategy at fixed tool orientation. The main objective was to find a feasible solution to the total problem rather than an optimal one. A fundamental requirement was that all approximations (of the objects involved and of the paths generated) should be conservative, *i.e.,* collisions and improper configurations should never occur, possibly at the expense of a nonoptimal process. This resulted in a method in which a weak coupling between the two aforementioned subproblems was maintained, as will be explained. The collision avoidance between the (relatively fast moving and wide-range) robot links and S has been solved recently and is reported here for the first time.

3. FINDING SUITABLE TOOL APPROACH ORIENTATIONS

The first of the two subproblems is the selection of a (preferably small) number of orientations for the tool T, which are to be applied later for the milling process. T is usually a rotation-symmetric tool, *e.g.* a cylinder with a spherical end. The axis of T has a direction relative to the reference frame in which the design model (or surface) F is defined. T is attached to (robot) R, which generally consists of several links in relative motion. There will be a part H of R which is always at rest relative to T. We will refer to H as the tool holder. Since H will in general not be rotation-symmetric we define the orientation $\mathbf{d}$ of H to be specified by the direction of T together with a twist angle. The goodness of an approach orientation depends on:1) the visibility from infinity of *finite* regions on the surface F, 2) the local curvature distribution of F, 3) the twist angle that maximizes the space for maneuver between H and F and 4) the (average) inclination angle of the tool relative to the local surface normal.

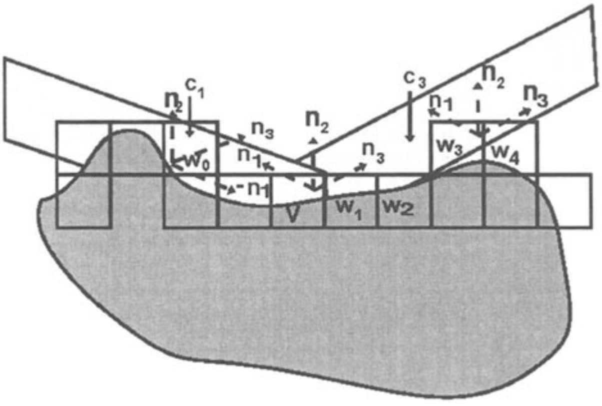

Figure 2. Obtaining the global accessibility directions for voxel v. For each of the directions $\mathbf{n}_i$ a beam of rays starting from the boundary of v into direction $\mathbf{n}_i$ is constructed. If the light map of any voxel w_i intersecting that beam contains a direction c_i opposite to $\mathbf{n}_i$ then $\boldsymbol{n}_i$ cannot be an accessibility direction for v.

The visibility of finite regions of F from a particular direction (implied by $\mathbf{d}$) corresponds to the accessibility of F by the finite tool T having that direction. The shape of each of the finite regions is the intersection of F with the cylindrical part of T, having its axis parallel to $\mathbf{d}$. This accessibility analysis is performed using a voxel approximation of F, where a global visibility map $V(v,F)$ and a light map $L(v,F)$ are assigned to each voxel v. $V(v,F)$ contains the directions of those rays starting from any point inside v on the surface F that do not intersect F any further. Each direction is defined by a pair of angles and can hence be represented by a point on the unit sphere. $V(v,F)$, which is thus a subset of the unit sphere, is approximated by a discrete set $V'(v,F)$ containing directions normal to the facets forming a dedicated tessellation of the unit sphere, called an icosahedron (Pugh 1976). $V'(v,F)$ is efficiently generated by first computing the *local* visibility map $V_l'(v,F)$ of v as

$$V_l'(v,F) = \bigcap_{\mathbf{p}\in F\cap v} V'(\mathbf{p},F),$$

where $V'(\mathbf{p},F)$ is the discrete local visibility map at point $\mathbf{p}$ on the surface of F, and approximately equals to the hemisphere defined by the surface normal at $\mathbf{p}$. If the points $\mathbf{p}$ in $F\cap v$ are sufficiently densely sampled on F then the set intersections generate a proper structure of local visibility voxels of F (Tangelder 1998). The light map $L(v,F)$ of v is the set of those directions in which the part of surface F enclosed by v is able to send radiation, taking into account the local geometry of F inside v, but not occlusion by remainder parts of F. $L(v,F)$ is constructed by taking the set union of the discrete hemispheres for points on $F\cap v$. The global visibility map $V'(v,F)$ of v is obtained by comparing its local visibility map $V_l'(v,F)$ to the light maps of those voxels from which v is visible (figure 2). Finally, the accessibility of a voxel v by the tool T is determined by the analysis of voxels in a neighborhood of v. In order to prevent the discretizations to introduce any overestimation of the accessibility, it is ensured that the visibility maps are conservative approximations, whereas the light maps are optimistic ones.

The computations are very efficient since the set intersections/unions of the various maps are implemented as bitwise AND/OR operations.

Depending on the morphology of F, there may exist boundary voxels which are inaccessible from any direction. The minimum number of directions needed to actually get access to the accessible voxels depends on the surface complexity as well, and typically ranges from 3 to 10 (figure 3).

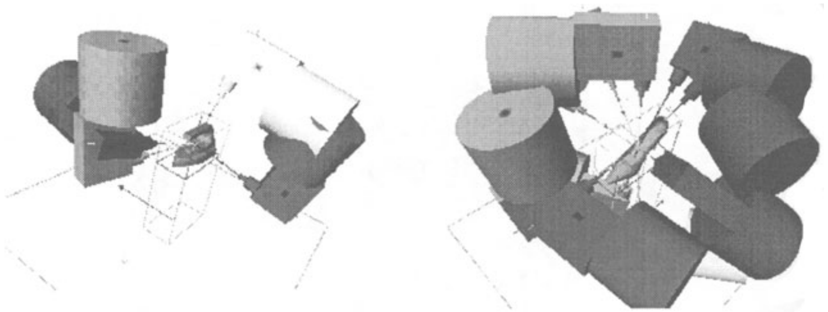

Figure 3. Two results from the tool approach orientation finder. 4 orientations are needed for the model of the iron (top); 10 orientations are required for the model of the shoe (bottom).

Whereas the suitable tool directions are derived exclusively from the geometry F, this is no longer the case for the orientations. The best twist angle, for a given access direction, is based (among other criteria) on the availability of a collision-free rectangular box near the stock. Later on this box will be used as a free space in which a machining path can be generated without having to test intersections between, *e.g.*, toolholder and turntable or between robot links and any obstacle. It will also be ensured that the free space remains within the workspace of the 7 DoF device. Collision detection is hence a precomputation, obviously mixed with the orientation selection process, which is against the separation principle we advocated in section 2. We therefore put two remarks. First, this mixing only occurs during the search for the best twist angle, not during the approach *direction* evaluation. Here we emphasize that, due to the many DoFs, and unlike in 5-axis machining, the twist angle can in principle take any value between 0 and 2π. Second, the computationally most expensive part of the collision avoidance deals with the following pairs of objects: 1) tool T

and design model F and 2) the toolholder H and the stock of material S. These pairs are *not* precomputed, but will be processed during the next phase, the path generation process itself.

4. PATH GENERATION IN THE PRECOMPUTED FREE SPACE

Based on the proposed tool approach orientations $\mathbf{d}_1, \ldots, \mathbf{d}_n$, the machining path computation is reduced to n local problems. The process starts with i=1 and calculates the occurring erosion S–T due to the movements of T having constant orientation $\mathbf{d}_i$. T can freely move inside the precomputed rectangular box (see section 3), except for potential (T,F) and (H,S) intersections. For the given $\mathbf{d}_i$ these intersections are tested using height maps of S and F and depth maps of H and T. The erosion of S is effected by continuously updating the S map. If the subprocess is exhausted, *i.e.*, no more material can still be removed at orientation $\mathbf{d}_i$, then i is incremented and the machining process is executed for the next orientation, until all n orientations have been applied. Due to the erosion of S, new maneuver space emerges. Therefore, the sequence of n orientations should be reattempted. Several rounds may be necessary before the whole process is completed, *i.e.* before no more material can be removed at any of the n orientations. To realize this process we needed to address a number of issues as follows.

Micro-level collision avoidance. For a given orientation $\mathbf{d}_i$ there is a continuous trade-off between, at the one hand, getting the tool at the lowest height (in a reference frame defined by $\mathbf{d}_i$) in order to maximize the (T,S) intersection and hence to remove as much material as possible, and at the other hand preventing collision between H and S. Optimally, H should slide just against S. Since S changes over time, this is a complex computation. We decided to maintain discrete z-maps of the four objects involved (F, S, H, T) at some spatial resolution d. Using specialized Minkowski operations on functions, it is facilitated to efficiently detect intersections and to perform continuous updating of the stock-in-progress (Tangelder, Vergeest & Overmars 1998). The resolution d can be used to control the quality of the surface finish.

Preservation of the consistency of the multiple height maps. If the machining at a particular orientation has finished then the z-map of S belonging to the n–1 other orientations should be updated in preparation of further path planning. This involves the intersection analysis of the n height maps in 3D space. Implicitly, the volume enclosed by these maps is a representation of F (Tangelder 1998).

Path planning strategy. As mentioned, the strategy to find a path for the tool inside the free space can be based on maximal depth of the tool as a function of the xy-position. This strategy is referred to as free space boundary following. Another strategy, which is in some cases more efficient, aims at moving at constant height, hence performing slicing. We compared both strategies using movements of the tool in zigzag patterns along the xy-plane. The slicing strategy appeared to perform up to four times better than boundary following, as far as machining time is considered (Tangelder 1998). In general it is favourable to minimize the number of transitions between successive straight path segments, as this will increase the average speed of the tool relative to S. This also provides a criterion for the choice between zigzagging in either x- or in y-direction.

5. MACROSCOPIC COLLISION AVOIDANCE

Special considerations were needed to prevent collisions between the mechanical device R as a whole and obstacles such as the turntable and the stock-in-progress S. In addition, there was the risk of collision *among* the links of R, which should be considered as well. We refer to these types of problems as macroscopic collision avoidance. We recall that the micro-level collision avoidance (in particular for the toolholder H) is treated differently, as described in section 4. We cannot use the micro-level methods for the macro-level situation since the objects involved (parts of R) have not a quasi-static orientation. Yet, complying to the principle of separation between collision avoidance and path planning, it has been attempted to bring the major part of the computation to the preprocessing stage. This has been achieved by replacing the precomputed rectangular free space boxes (one for each tool orientation, as described in section 3) by boxes having arbitrarily shaped bottom faces, to account for collisions of parts of R. The bottom face is represented by a discrete z-map, constraining the tool paths during the planning stage (Kovács 1998).

The main issue, of course, is to actually obtain the lower z-map of the free space. For a given grid in the xy-plane the new boundary positions are found by letting T step down along the z-direction until for the assumed configuration of R a collision is detected. To efficiently perform the collision detection, geometrical approximations of R and of S have been devised in such a way that 1) the volume specified by the approximation strictly encloses the original volume in order to avoid undetected collisions and 2) the intersection calculations are fast. For this last reason it is mandatory to accelerate the fundamental geometric check. This issue was addressed, *e.g.*, by (Benchetrit, Lenz & Shoham 1998). They based their solution on an approximation of the objects using a hierarchical representation of bounding boxes. The fundamental check is then the intersection calculation of bounding boxes.

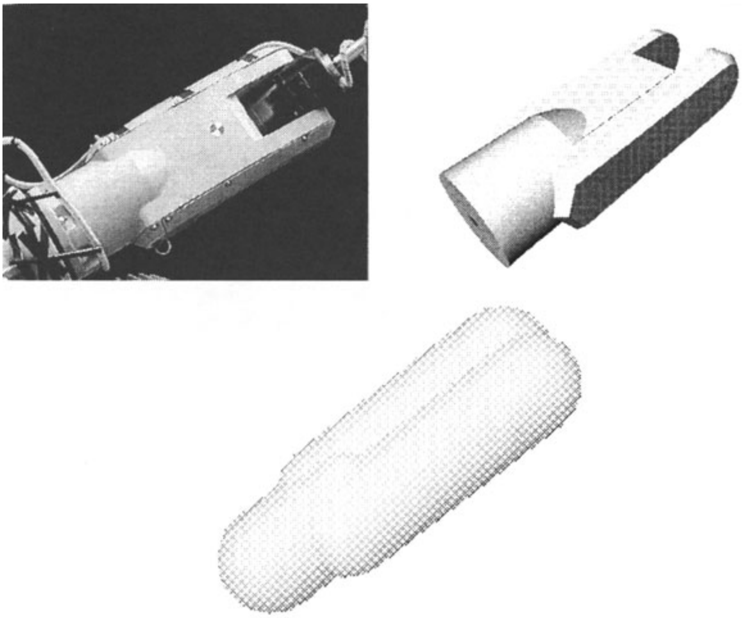

Figure 4. Each link of the robot R, is at first available as a B-spline surface model. Then it is approximated by a number of spheres and cylinders.

We have used spheres and cylinders as the building entities for the approximations. Collision detection has hence been reduced mainly to the calculation of distances between points (Kuczogi 1998). The geometric precision of the approximation can be controlled and influences the number of entities needed to represent the objects. Here the trade-off is between computation effort and the risk of false hits. The best balance has been achieved by using relatively large spheres for the mechanism R and smaller ones for S (figures 4 and 5).

The collision detection is performed in a preprocess; the tooltip is lowered (figure 5) and sphere/sphere and sphere/cylinder intersection calculations determine the free space range for given xy-grid (figure 6). Thus the actual free space is identified.

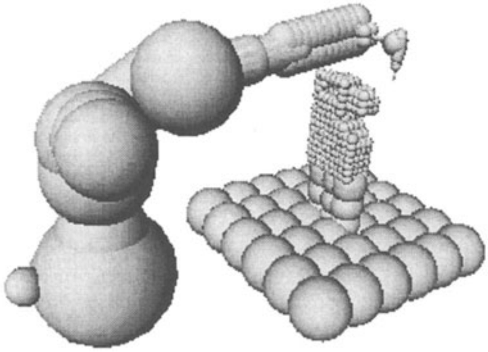

Figure 5. Sphere approximation of robot, turntable and stock-in-progress.

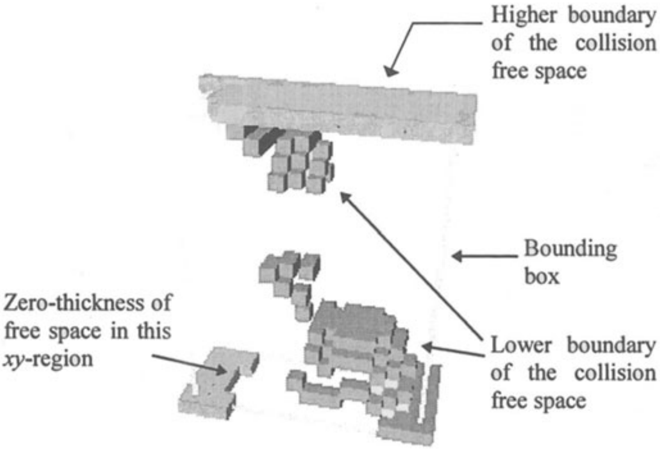

Figure 6. For each xy-grid it is determined which range in the z-direction belongs to the free space. The right part of the bounding box has a wide range; the leftmost part of the box has coinciding higher and lower boundary, and hence zero-range free space.

6. PRACTICAL RESULTS AND REMAINING ISSUES

The purpose of the research was to explore the feasibility of rapid prototyping using the Sculpturing Robot. To this end, the machining method just described has been implemented in software, called MAOS, in a C++ programming environment. Emphasis was directed to the support of conceptual shape design. This made it necessary for the system to be rather tolerant with regard to the imported 3D model F. In particular the system is able to accept models that only roughly or even incompletely define a volume; cracks and extrusions below a (user-defined) size are ignored and the intended solid is reconstructed from the inaccurate data (Tangelder 1998).

What the user (the concept designer) should specify is the required spatial accuracy for the machining process. This obviously influences the speed of computation and fabrication. Finally, to support the optimal selection of the machining accuracy and to increase the effectiveness of the RP facility, there is an option to preview the entire machining process using a dedicated simulation software package called SRSIM (Walstra, Bronsvoort & Vergeest 1994). The eroding workpiece can be visualized either in real-time or accelerated, at randomly selected points in time of the process.

Figure 7 presents a snapshot from the simulation and of the physical process of the fabrication of a CAD-defined model. The design was the housing for a car mirror, approximately 24 cm long. Evaluation tests involving different types of 3D models have revealed that:

- The orientation selection functions properly; those regions on the object's surface that are beam-wise visible are actually identified. Mostly, this results in a limited number of approach orientations (typically 5) for the tool.
- For each of these orientations an efficient machining strategy can be found.
- Both micro- and macro-level collision avoidance is feasible.
- In light plastic material, an average path spacing of about 1 mm already provided sufficient surface quality.

Typically, the path planning computation takes 1 to 5 hours (on a medium-cost Unix workstation), and the machining duration is between 3 and 10 hours. We emphasize that the machining itself is a fully automatic process without refixturing or other manual intervention.

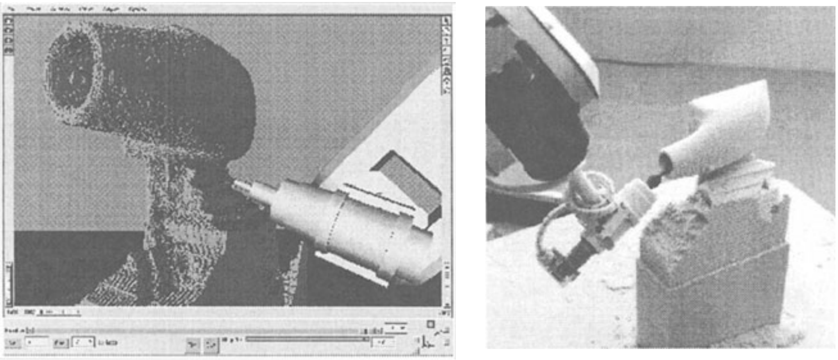

Figure 7. The entire machining process can be previewed (left). The Sculpturing Robot in operation is shown in the picture to the right.

The most challenging issues are the further acceleration of the computation, both in the search for optimal approach orientations and in the calculation of the machining path. Furthermore, it is expected that the machining time can be significantly reduced. A significant drawback of the present SR system is its dependence on the reachability by the tool of the designed surface; particularly, inner structures of the design are generally unreachable. Another limitation is the finite workspace of the device, which puts bounds to the dimensions of the produced part. To cope with these issues it will be necessary to automatically decompose the imported model into fully accessible portions, which can hence be machined and finally assembled. One current approach to this so-called hybrid rapid prototyping is the fast fabrication of thick slices using flexible blade cutting (Horváth, Vergeest & Juhász 1998).

7. CONCLUSIONS

We have presented a computationally feasible method for the fully automatic planning of machining paths for the fabrication of arbitrarily complex shapes. By separating the search for tool orientations and for tool paths, an appropriate balance was achieved of such aspects as surface quality, collision avoidance, exploitation of the high-dimensional configuration space and efficient tool motion planning. We have successfully evaluated the method using a hardware/software implementation, called the Sculpturing Robot, which is now in operation as a rapid prototyping facility.

ACKNOWLEDGMENTS

The authors are very much indebted to their colleagues Bram de Smit, Han Broek and Adrie Kooijman. Many thanks to Mark Overmars from Utrecht University for his very important scientific support.

REFERENCES

1. U. Benchetrit, E. Lenz and M. Shoham (1998), "Collision detection in static and dynamic complex scenes not having geometry coherence". Int. J. Advanced Manufacturing, Vol. 14, pp 70-76.1
2. B. Choi, J. Park and C. Jun (1993), "Cutter-location data optimization in 5-axis surface machining". Computer-Aided Design, Vol. 29, Nr. 10, pp 709-725.
3. I. Horváth, J.S.M. Vergeest, J.J. Broek, and A. de Smit (1998), "Tool Profile and Tool Path Calculation for Free-form Thick-layered Fabrication". To appear in Computer-Aided Design.
4. Z. Kovács (1998), "Integrating the free space generator with the sculpturing robot software". Technical report, Delft University of Technology, Delft.
5. Gy. Kuczogi (1998), "Collision avoidance for the sculpturing robot". Technical report, Delft University of Technology, Delft.
6. A. Pugh (1976), "Polyhedra: A visual approach". Springer-Verlag, London.
7. J.W.H. Tangelder (1998), "Automated fabrication of shape models of free-form objects with a sculpturing robot". PhD thesis, Delft University of Technology, Delft.
8. J.W.H. Tangelder, J.S.M. Vergeest and M. Overmars (1998)"Interference-free NC machining using spatial planning and Minkowski operations".Journal of Computer-aided Design, Vol. 30, Nr. 4, pp 277-286.
9. J.S.M. Vergeest and J.W.H. Tangelder (1996), "Robot machines rapid prototype". Industrial Robot, Vol. 23, Nr. 5, pp 17-20.
10. J.S.M. Vergeest, J.J. Broek and J.W.H. Tangelder (1995) "Rapid prototyping of largecomplex shapes with a fully automatic robot system". Proc. 5th Int. Conf. on Computer Applications in Production and Engineering, CAPE'95, (Q.N. Sun, Ed.), IFIP, Chapman & Hall, London, pp 773-780.
11. M. Wall, K. Ulrich and W. Flowers (1992), "Evaluating prototyping technologies for product design". Research in Engineering, Vol. 3, pp 163-177.
12. W.H. Walstra, W.F. Bronsvoort and J.S.M. Vergeest (1994), "Interactive simulation of robot milling for rapid shape prototyping. Computers & Graphics, Vol. 18, No. 6, pp 861-871.
13. C.-F. You and C.-H. Chu (1997), "Tool-path verification in five-axis machining of sculptured surfaces". Int. Journal of Advanced Manufacturing Technology, Vol. 13, pp 248-255.