

Parting directions for mould and die design

Lin-Lin Chen, Shuo-Yan Chou* and Tony C Woo†

On the basis of the condition for demouldability, two levels of visibility, *complete* and *partial* visibility, are defined. The viewing directions from which a surface is completely visible can be represented as a convex region on the unit sphere called the *visibility map* of the surface. Algorithms are given for dividing a given object into pockets, for which visibility and demouldability can be determined independently, for constructing visibility maps, and for selecting an optimal pair of parting directions for a mould that minimizes the number of cores. An example illustrates the algorithms.

Keywords: manufacturing, algorithms, computational geometry, injection moulding, die casting, parting directions, visibility

In die casting and injection moulding¹⁻⁴, a basic mould consists of two plates that form an impression into which molten material is injected. The surfaces on the two plates that meet to form a seal when the mould closes are the *parting surfaces*; the pair of opposite directions along which the two plates of the mould separate are the *parting directions* (see *Figure 1*). Recesses or projections on the moulded piece that prevent its removal from the mould along the parting directions are called *undercuts*. Depending on the types of undercut, different manufacturing devices are used: external recesses and projections are formed by using *side cores* and *side cavities*; internal undercuts are formed by using *mould pins* or *inserts*. An example of an object requiring a side core for an external recess is shown in *Figure 2*. Here

and after, the general term *core* is used to refer to a side cavity, a side core, a mould pin, or an insert.

The selection of the parting directions and parting surfaces is important, because it dictates the number and the shapes of cores, and affects all the subsequent steps in the design of a mould. Because the use of cores increases tooling costs, complicates the operation of the mould, and slows down the process, a general rule is emphasized in the literature^{1,2,5} that parting directions should be selected so that the number of cores is kept to a minimum. However, the automation of this step has received little attention in the past. Of the few examinations reported, the parting surface is restricted to a plane by assuming that the parting direction is along one of the three principal axes⁶⁻⁸, or selected from a set of randomly generated directions⁹. A drawback of these approaches is the impossibility of knowing whether a feasible parting direction exists at all, and, if more than one exists, whether one is better than the others. Heuristically generated candidate parting directions that consist of the normals of planar faces and the axes of cylindrical surfaces of the given object are also reported¹⁰. The feasibility of a parting direction is verified by checking sample points on the object for obstruction in the candidate direction.

In this paper, a deterministic method is developed for finding an optimal pair of parting directions. Given the geometry of an object, depending on the selection of the pair of parting directions, a different number of cores

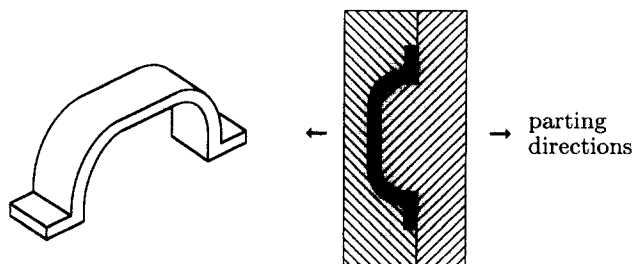


Figure 1 Example of mould and corresponding parting directions

Institute of Engineering Technology, National Taiwan Institute of Technology, 43 Keelung Road Section 4, Taipei, Taiwan

* Department of Industrial Management, National Taiwan Institute of Technology, 43 Keelung Road Section 4, Taipei, Taiwan

† Department of Industrial and Operations Engineering, University of Michigan, 10E Building, 1205 Beal Avenue, Ann Arbor, MI 48109-2117, USA

Paper received: 22 August 1992. Revised: 15 March 1993

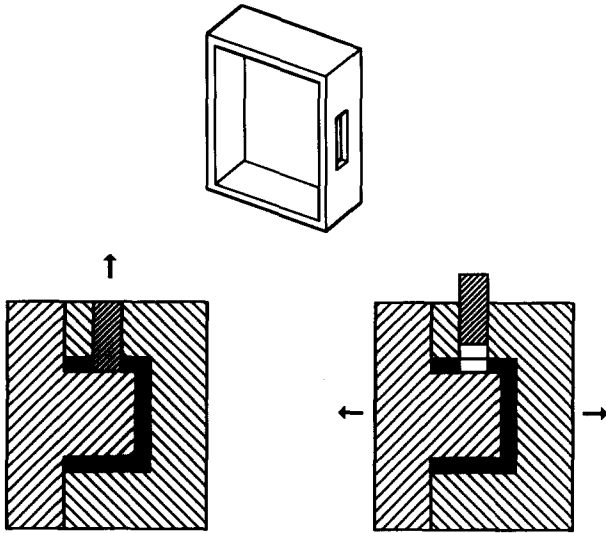


Figure 2 Example of object requiring side core for external recess

may be required. 'Optimal' denotes minimizing the number of required cores.

Problem PPD (pair of parting directions): Given an object, find a pair of opposite parting directions that minimizes the number of cores.

Basic to problem PPD is the condition for demouldability. A surface is demouldable along a certain direction if the entire surface does not contain any undercut. This condition is satisfied if the entire surface can be illuminated by parallel rays in the parting direction, i.e. if the entire surface is *visible* to parallel rays in the parting directions. Cores are required for the portions not illuminated by the parallel rays. The second section further defines the relation between demouldability and visibility, and discusses the domain of objects covered. With the development of visibility maps on the Gaussian sphere, the third section transforms problem PPD to the maximal covering of spherical polygons by a point, followed by an algorithm. The fourth section illustrates the algorithm with an example.

VISIBILITY AND DEMOULDABILITY

Given an object Ω , a point p on the boundary of Ω is *visible* to an exterior point q if no part of the line segment pq lies in the interior^{11,12} of Ω . Extending the notion of point visibility, a surface S on Ω is *completely visible* to an exterior point q if every point on S is visible to q ; S is *partially visible* to q if at least one point on S is visible to q ; S is *not visible* to q if no point on S is visible to q . The visibility of a surface in a viewing direction can then be defined through a limiting process. As the point q is moved away from S towards infinity, line segments connecting points on S and the point q approach being

parallel. Indeed, in projective geometry^{13,14}, a viewing direction \mathbf{d} is a point at infinity, and rays reaching points on S in the direction \mathbf{d} are line segments connecting the point \mathbf{d} at infinity and points on the surface. Thus, a surface has two levels of visibility with respect to a viewing direction.

Definition 1 (complete visibility): A surface S on a polyhedral object Ω is completely visible in a viewing direction \mathbf{d} if, for every point p on S , the ray starting from infinity to p in the direction \mathbf{d} does not intersect the interior of Ω .

Definition 2 (partial visibility): A surface S on a polyhedral object Ω is partially visible in a viewing direction \mathbf{d} if there exists at least one point p on S such that the ray from infinity to p in the direction \mathbf{d} does not intersect the interior of Ω .

These two levels of visibility are illustrated in *Figures 3 and 4*. By definition, if a surface is completely visible in a viewing direction, then it is also partially visible in the same direction.

The condition for demouldability can now be formulated by examining the trajectory formed by a surface upon its removal from the mould. When a surface S is removed from the mould along a parting direction \mathbf{d} , the trajectory of each point on S forms a ray from the point to infinity in the direction \mathbf{d} . Let \mathcal{R} denote the set of all rays formed by the trajectories of points on S . Then, S is removable from the mould if none of the rays in \mathcal{R} intersects the mould. This condition for the demouldability of a surface S along a parting direction \mathbf{d} coincides with the condition for the complete visibility of S in a viewing direction $-\mathbf{d}$. Thus, given the surfaces on the object, and, for each surface, the corresponding set of viewing directions from which the surface is completely visible, problem PPD can be solved by selecting a pair of opposite directions that



Figure 3 Complete visibility and demouldability

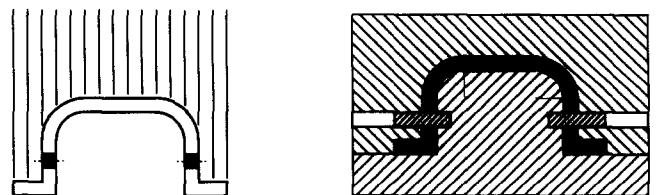


Figure 4 Partial visibility and use of cores

maximizes the number of surfaces that are completely visible from the pair of viewing directions.

The complete visibility of a surface can be destroyed owing to *local interference*, parts of the same surface, or *global interference* by other surfaces of the object. The set of viewing directions in which the visibility of a surface is free of any local interference can be computed and represented as a spherically convex region called the *visibility map* of the surface.

Local interference and visibility map

A surface can be mapped onto the unit sphere by translating the normal at every point of the surface to the origin, and then intersecting it with a unit sphere centred at the origin. This process, developed by Gauss, is called *Gaussian mapping*, and the spherical representation of the surface thus obtained is called the *Gaussian map* (or *GMap*) of the surface¹⁵. The GMap of a polyhedron consists of a finite number of spherical points, whereas the GMap of a curved surface is, in general, a spherical region.

Locally, a point on a surface is visible from more than a single viewing direction. Let \mathbf{n} be the normal and T be the tangent plane at a point p on a surface S . The point p is visible from all the directions up to a hemisphere, with \mathbf{n} being at the 'north pole' and T being the 'equatorial' plane. As there are many points on the surface, the intersection of these hemispheres consists of all the directions from which the entire surface is locally visible. The resulting spherically convex region is called the *visibility map* (or *VMap*) of the surface. Any point in the VMap corresponds to a direction from which the entire surface is locally visible. Examples of surfaces and their corresponding GMaps and VMaps are shown in *Figure 5*. It is intuitive that the more 'complicated' the surface is, the larger the GMap is and the smaller the VMap is. This 'inverse' relation between the two maps is succinctly illustrated by the point-hemisphere duality shown in the first two rows of *Figure 5*. It may also be noted that the VMap may be empty, as shown in the last row of *Figure 5*. The VMap of a surface can be computed by using the $O(n \log n)$ time algorithm given in Reference 16 for intersecting a set of n hemispheres corresponding to n sample points on a surface.

Global interference and pockets

It is self-evident that the surfaces of a convex object suffer no global interference. However, not all objects and, in particular, surfaces are convex. The notion of a *pocket* is useful for detecting global interference. Let $\text{CH}(\Omega)$ denote the convex hull of an object Ω . If a surface S of Ω also belongs to $\text{CH}(\Omega)$, then S is completely visible. Let $\bar{P}_1, \bar{P}_2, \dots, \bar{P}_m$ be the set of polyhedra that result from taking the regularized difference¹⁷ between $\text{CH}(\Omega)$ and Ω ,

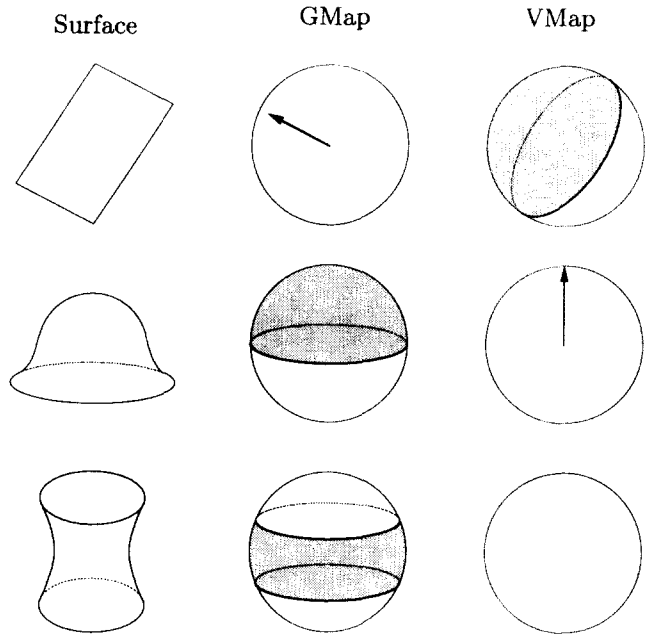


Figure 5 Examples of surfaces and their corresponding GMaps and VMaps

denoted by $\text{CH}(\Omega) - * \Omega$. Each polyhedron \bar{P}_i is called a *sealed pocket* of Ω , and consists of two types of surfaces: those that belong to $\text{CH}(\Omega)$ but not to Ω , and *vice versa*. The former are called the *lid surfaces*, as they form *lids*[†] that seal the *pocket*, which is formed by the latter type of surface, called *pocket surfaces*. Let $\text{Lid}(\bar{P}_i)$ and $\text{Pocket}(\bar{P}_i)$ denote the lid surfaces and the pocket surfaces of \bar{P}_i . An example of an object, its convex hull, the corresponding sealed pockets, and pockets are given, respectively, in *Figures 6a-d*. Effectively, a single object is decomposed into disjoint subsets, each of which is amenable to processing for visibility. The fact that the visibility of the points in a pocket can be considered independently of that of the other pockets is now considered.

Lemma 1: If the visibility of a point p in a sealed pocket \bar{P}_i in the viewing direction \mathbf{d} is not interfered with by any surface in $\text{Pocket}(\bar{P}_i)$, then the point p is visible in the viewing direction \mathbf{d} .

Proof: Let q be the first intersection point of the ray that is emitted from the point p in the direction $-\mathbf{d}$ with surfaces of the sealed pockets $\bar{P}_1, \bar{P}_2, \dots, \bar{P}_m$. The point q must lie on a surface of \bar{P}_i , since, if q is on a face of $\bar{P}_j, j \neq i$, then a line segment L can be constructed such that L is a subset of $\text{CH}(\Omega) - * \Omega$ and connects \bar{P}_i and \bar{P}_j . This contradicts the fact that \bar{P}_i and \bar{P}_j are disjoint.

If q lies on a surface S in $\text{Pocket}(\bar{P}_i)$, then the visibility of p is blocked by the pocket surface S . Otherwise, if q lies on a surface in $\text{Lid}(\bar{P}_i)$, then, by construction, q is

[†] It is noted that a pocket may require more than one lid to be sealed off.

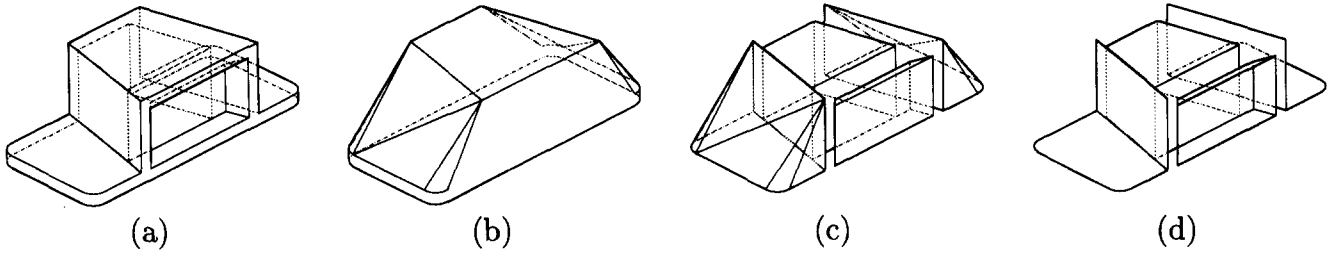


Figure 6 Object; (a) object, (b) its convex hull, (c) corresponding sealed pockets, (d) pockets

visible in the viewing direction \mathbf{d} . Thus, p is also visible in the viewing direction \mathbf{d} . \square

When Ω is a polyhedral object with n faces, its convex hull $CH(\Omega)$ can be computed in $O(n \log n)$ time^{18,19}, and the regularized difference between $CH(\Omega)$ and Ω can be determined in $O(n \log n)$ time²⁰. Hence, the set of sealed pockets can be identified in $O(n \log n)$ time. The set of pockets can be identified in an additional $O(n)$ time. The set of directions from which a pocket is completely visible and thus demouldable is the VMap of the surface, which can be computed as described in the section above on visibility maps.

PARTING DIRECTIONS

Problem PPD seeks a pair of opposite directions such that the number of cores is minimized. As described previously, a set of pockets $\mathcal{P} = P_1, P_2, \dots, P_m$ can be extracted from Ω , where $P_i = \text{Pocket}(\bar{P}_i)$. Let $\mathcal{VM} = \{VM(P_1), VM(P_2), \dots, VM(P_m)\}$ denote the corresponding visibility maps, which are assumed to be nonempty.

For a pair of opposite directions \mathbf{d} and $-\mathbf{d}$, \mathcal{P} can be divided into three subsets, \mathcal{P}^+ , \mathcal{P}^- and \mathcal{P}^0 , which consist of those surfaces completely visible from \mathbf{d} , those completely visible from $-\mathbf{d}$, and those not completely visible from either \mathbf{d} or $-\mathbf{d}$, such that

$$\mathbf{d} \in \bigcap_{P_i \in \mathcal{P}^+} VM(P_i)$$

and

$$-\mathbf{d} \in \bigcap_{P_i \in \mathcal{P}^-} VM(P_i)$$

With \mathbf{d} and $-\mathbf{d}$ as parting directions, surfaces in \mathcal{P}^+ and \mathcal{P}^- can be incorporated into the mould, while surfaces in \mathcal{P}^0 require cores. This suggests a way of determining the number of cores, by counting the number of surfaces in \mathcal{P}^0 .

Since the VMaps are spherically convex polygons, and a pair of opposite directions can be represented as antipodal points that are diagonally opposed*, problem

*Two spherical points p and q are called antipodal if $q = -p$. The point q is called the antipode of p , and vice versa.

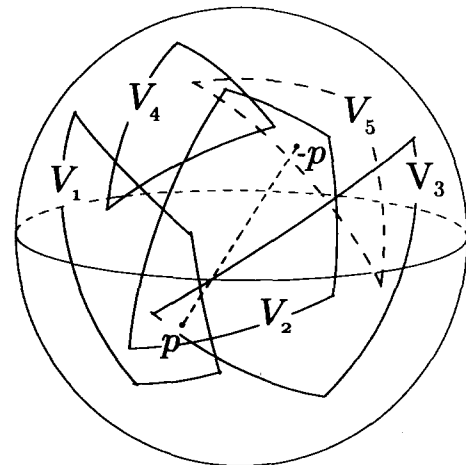


Figure 7 Spherical polygon covering by antipodal points

PPD can be formulated as a spherical polygon covering the problem. Figure 7 shows five spherically convex polygons V_1, \dots, V_5 and a pair of antipodal points p and $-p$. As any point p in the intersection of the spherical polygons (e.g. $V_1 \cap V_2 \cap V_3$) denotes a direction in which the corresponding surfaces (e.g. S_1, S_2 and S_3) are completely visible, problem PPD becomes one of locating a pair of antipodal points such that they are contained in (or covered by) the maximal number of VMaps. The surfaces corresponding to VMaps not covered by the pair of points would then require cores.

Problem (spherical polygon covering by antipodes): Given a set of spherically convex polygons V_1, V_2, \dots, V_m , find a pair of antipodal points p and $-p$ that maximize the number of V_i containing either p or $-p$.

A pair of antipodes may be thought of as a single point if the polygons can somehow be 'mirrored' spherically. Let \mathcal{V} denote the given set of spherically convex polygons. A duplicate set of spherically convex polygons $-\mathcal{V}$ consisting of polygons opposite† to those in \mathcal{V} is introduced. If a point p lies in the common intersection of the spherically convex polygons V_i and $-V_j$, then the

† Let V be a k -sided spherical polygon with vertices p_1, p_2, \dots, p_k , in that order. Then, the opposite of V is another spherical polygon $-V$ with vertices q_1, q_2, \dots, q_k , in that order, where q_i is the antipode of p_i .

point p lies in V_i , and its antipode $-p$ lies in V_j . This observation leads to an alternative formulation of problem PPD, as follows.

Problem (spherical polygon covering by a point): Given a set of spherically convex polygons $V_1, V_2, \dots, V_m, -V_1, -V_2, \dots, -V_m$, find a point p that maximizes the number of polygons containing p .

Thus, problem PPD can be solved by finding a maximally covered point, through a spherical partition induced by the given set of polygons.

Each point p on the sphere can be assigned an ownership vector $u(p)$, where

$$u(p) = (u_1(p), u_2(p), \dots, u_m(p))$$

and

$$u_i(p) = \begin{cases} 1 & p \in V_i \\ -1 & p \in -V_i \\ 0 & \text{otherwise} \end{cases}$$

The vector $u(p)$ thus keeps track of the polygons covering p . Define two points p and q to be equivalent if $u(p) = u(q)$. Then, a cell in the partition is the largest 2D connected subset of equivalent points. Let the ownership vector of a cell K be the same as that of any point in K , i.e.

$$u(K) = u(p)$$

for any $p \in K$.

Two adjacent cells and their ownership vectors are shown in Figure 8. It is observed that the ownership vectors of two adjacent cells differ by exactly one element: the polygon whose edges separate the two cells. Thus, given the ownership vector of a cell, the ownership vector of the other cells can be obtained through propagation by using the adjacency relations among them.

Since the polygons covering a cell are recorded by its ownership vector, a maximally covered point can be

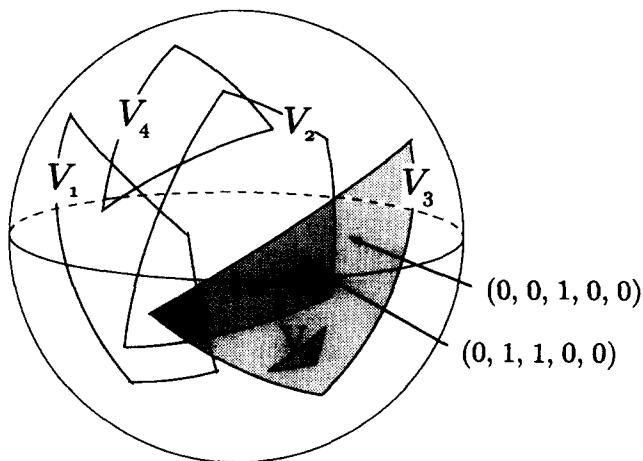


Figure 8 Two adjacent cells and their ownership vectors

found by traversing the cells in the partition and selecting a point in a cell K that maximizes the value $|u(K)|$, where

$$|u(K)| = \sum_{i=1}^m |u_i(K)|$$

This leads to the following algorithm:

Algorithm (spherical polygon covering by a point)

Input: A set of polygons $\mathcal{V} = \{V_1, \dots, V_m, -V_1, \dots, -V_m\}$

Output: A point p that maximizes the number of polygons containing it

- (1) Compute the spherical partition induced by the polygons in \mathcal{V} .
- (2) Select a maximally covered point p :
 Compute the ownership vector $u(K_0)$ for an arbitrary cell K_0 .
 $max_vector \leftarrow u(K_0), max_covered_cell \leftarrow K_0$
 $K \leftarrow K_0, u(K) \leftarrow u(K_0)$
 Perform a depth-first search of the partition.
for each cell $\tilde{K} \neq K_0$ do
 Compute $u(\tilde{K})$ by updating $u(K)$.
 if $|max_vector| < |u(\tilde{K})|$ then
 $max_vector \leftarrow u(\tilde{K}),$
 $max_covered_cell \leftarrow \tilde{K}$
 $K \leftarrow \tilde{K}, u(K) \leftarrow u(\tilde{K})$
 $p \leftarrow$ a point in $max_covered_cell$
Output p .

Let n_i denote the number of vertices of the spherically convex polygon V_i and let $n = \sum_{i=1}^m n_i$. Then, Steps 1 and 2 require $O(nm \log m)$ time and $O(nm)$ time, respectively. Thus, the time complexity of the spherical polygon cover by a point algorithm is $O(nm \log m)$.

EXAMPLE

The selection of parting directions using the spherical polygon covering by a point algorithm is shown with the object in Figure 6. Figure 9 shows the four pockets S_1, S_2, S_3 and S_4 of the object and their corresponding VMaps V_1, V_2, V_3 and V_4 , where V_1 and V_2 are quadrants of the Gaussian sphere, V_3 is a spherical rectangle, and V_4 consists of a single point at the south pole. The spherical partition induced by the polygons $V_1, V_2, V_3, V_4, -V_1, -V_2, -V_3,$ and $-V_4$ is shown in Figure 10, where d is a maximally covered point that lies in V_1, V_2 and $-V_4$. Choosing d and its opposite $-d$ as parting directions, S_1 and S_2 are completely visible from d , and S_4 is completely visible from $-d$. Thus, S_1, S_2 and S_4 can be incorporated into the mould, while S_3 requires the use of a core. In Figure 10, another maximally covered point is d' , which lies in $-V_1, -V_2$ and V_3 . By choosing d' and its opposite $-d'$ as parting directions, the surfaces S_1, S_2 and S_3 can be incorporated into the mould, whereas the surface S_4 requires the use of a core. The spherical

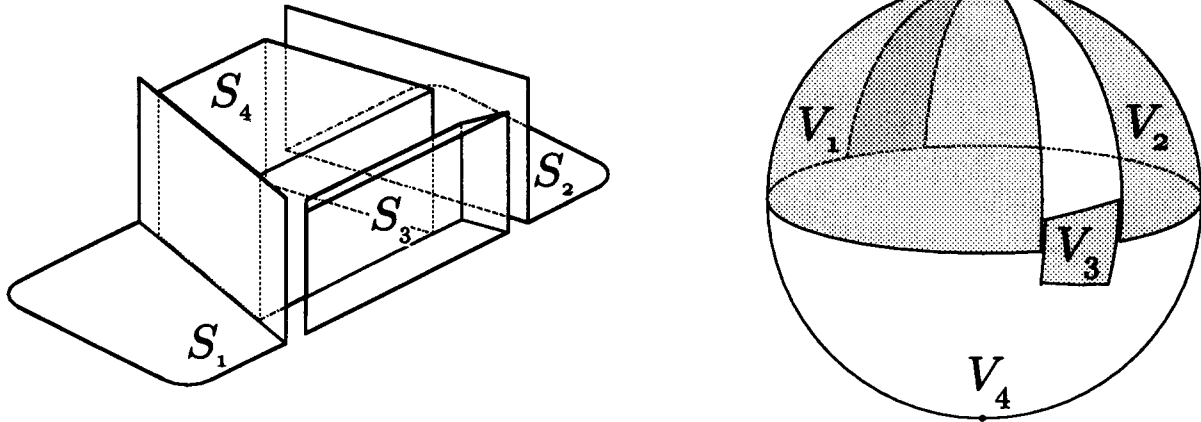


Figure 9 Pockets of object in Figure 6 and their corresponding VMaps

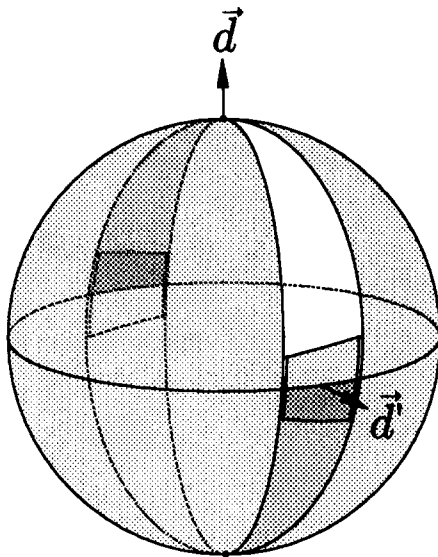


Figure 10 Spherical partition induced by VMaps in Figure 9 and their opposites

polygon covering by a point algorithm reports the first maximally covered point that it encounters during the traversal of the spherical partition.

CONCLUSIONS

An $O(nm \log m)$ time algorithm has been developed for determining optimal parting directions for an object with m pockets and n vertices.

During the development of the algorithm, it is assumed that all the cores are equally undesirable. Not explicitly stated is the assignment of a weight w_i to a pocket S_i on the basis of its geometric complexity or an estimate of its manufacturing cost. The algorithm can be modified such that, instead of minimizing, over all possible directions \mathbf{d} , the value $|u(\mathbf{d})|$ which is equal to the sum of the individual $u_i(\mathbf{d})$, the objective is to minimize the

weighted sum

$$\sum_{i=1, \dots, n} u_i(\mathbf{d})w_i$$

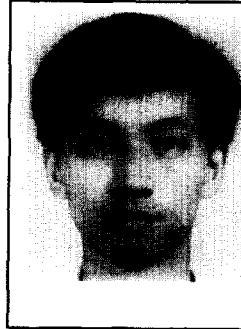
The time complexity is not affected by this modification.

It is also assumed that a core is used for a pocket that has an empty VMap, and is thus not completely visible from any direction. However, by subdividing a pocket, the use of cores can sometimes be completely eliminated. Algorithms that consider the subdivisions of a pocket by invoking the notion of *partial visibility* are currently being developed.

REFERENCES

- 1 Bralla, J G (Ed.) *Handbook of Product Design for Manufacturing: A Practical Guide to Low-Cost Production* McGraw-Hill (1986)
- 2 Campbell, J *Castings* Butterworth-Heinemann (1991)
- 3 Pye, R G W *Injection Mold Design: A Textbook for the Novice and a Design Manual for the Thermoplastics Industry* (4th Ed.) John Wiley (1989)
- 4 Rubin, I I *Handbook of Plastic Materials and Technology* John Wiley (1990)
- 5 Kalpakjian, S *Manufacturing Processes for Engineering Materials* Addison-Wesley (1991)
- 6 Lee, K and Smith, C T 'Computer-aided pattern design for casting processes' *Manufacturing Processes, Systems & Machines Proc. 14th Conf. Production Research & Technology* (1987) pp 365-371
- 7 Ravi, B and Srinivasan, M N 'Decision criteria for computer-aided parting surface design' *Comput.-Aided Des.* Vol 22 No 1 (1990) pp 11-18
- 8 Smith, C T and Lee, K 'Computer-aided pattern design for casting processes' *AFS Trans.* (1986) pp 21-28
- 9 Ganter, M A and Tuss, L L 'Computer-assisted parting line development for cast pattern production' *AFS Trans.* (1990) pp 795-800
- 10 Hui, K C and Tan, S T 'Mould design with sweep operations - a heuristic search approach' *Comput.-Aided Des.* Vol 24 No 2 (1992) pp 81-91
- 11 O'Rourke, J *Art Gallery Theorems and Algorithms* Oxford University Press (1987)
- 12 Toussaint, G T 'Computing visibility properties of polygons' in *Pattern Recognition and Artificial Intelligence* North-Holland (1988)
- 13 Stolfi, J *Oriented Projective Geometry: A Framework for Geometric Computations* Academic Press (1991)

- 14 Struik, D J *Lectures on Analytic and Projective Geometry* Addison-Wesley (1953)
- 15 Hilbert, D and Cohn-Vossen, S *Geometry and the Imagination* Chelsea (1952)
- 16 Chen, L L and Woo, T C 'Computational geometry on the sphere for automated machining' *ASME J. Mech. Des.* Vol 114 No 2 (1992) pp 288-295
- 17 Hoffmann, C M *Geometric and Solid Modeling: An Introduction* Morgan Kaufmann (1989)
- 18 Edelsbrunner, H *Algorithms in Combinatorial Geometry* Springer-Verlag (1987)
- 19 Preparata, F P and Hong, S J 'Convex hulls of finite sets of points in two and three dimensions' *Commun. ACM* Vol 2 No 20 (1977) pp 87-93
- 20 Tang, K and Woo, T C 'Algorithmic aspects of alternating sums of volumes. Part I: Data structure and difference operation' *Comput.-Aided Des.* Vol 23 No 5 (1991) pp 357-366

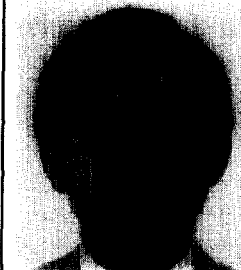


Shuo-Yan Chou is an associate professor of engineering management. His research interests are in computational geometry and its applications to problems that arise in design, process planning, manufacturing, and inspection. He received a BBA in industrial management science from Cheng-Kung University, Taiwan, in 1983, and an MS and a PhD in industrial and operations engineering from the University of Michigan, USA, in 1987 and 1992.



Lin-Lin Chen is an associate professor of engineering technology. Her research interests are in design for manufacturing, computational geometry, solid modelling, and geometric tolerancing. She received a BS in industrial design from Cheng-Kung University, Taiwan, in 1984, and a PhD in industrial and operations engineering from the University of Michigan, USA, in 1992. She worked as a design engineer in Sampo Corporation, Taiwan, from 1984 to 1986, and as a

half-time computer programmer in the computer-aided engineering department of the Ford Motor Company from 1987 to 1989.



Tony C Woo is a professor of industrial and operations engineering and a professor of mechanical engineering and applied mechanics, and he teaches courses on computer graphics, computer-aided design, and geometric modelling. His research interest is in the development of efficient data structures and algorithms for processing geometry. His training was in electrical engineering. He is also a consultant to industries. Between 1987 and 1990, he served at the US National

Science Foundation, and in 1991, he was a Fulbright Scholar at NTH in Trondheim, Norway. In 1992/93, he is a visiting professor with the Computer Science Department at Johns Hopkins University, USA.