Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation
July 16-20, 2003, Kobe, Japan

Part Geometric Understanding for Tool Path Planning in Additive Manufacturing

Weihua Sheng
ECE. Department.
Kettering University
Flint, MI 48504

Ning Xi, Heping Chen
ECE. Department.
Michigan State University
East Lansing, MI 48824

Yifan Chen
Scientific Research Lab
Ford Motor Company
Dearborn, MI 48121

Mumin Song
Control Science and Engineering Dept.
Shandong University
Shandong, China

Abstract

Additive manufacturing processes such as spray coating, spray painting and rapid tooling, are very important steps in many products' life cycle. Robotic manipulators are widely adopted in these processes. The tool planning for these applications is usually time-consuming for human operators and the plans generated are prone to inaccuracy and low efficiency. This research develops a fully-automated, CAD-guided tool planning system which eliminates the human involvement and also generates optimized tool plans in the sense of the motion performance of the robot. The core of this tool planning system is the partitioning of part surfaces into multiple, easy-to-handle patches, or the geometric understanding problem. In this paper, a decomposition-based approach is developed. Robot motion performance measures are integrated into the surface partitioning. Experimental tests and evaluation carried out on vehicle body parts validate this new approach.

1 Introduction

Additive manufacturing is a very important step in many products' life cycle. Typical additive manufacturing processes include spray coating, spray painting, metal deposition for rapid tooling. Additive manufacturing also plays a key role in improving products' appearance and quality, such as furniture and home appliances. More and more additive manufacturing processes are carried out by robotic manipulators with specific tools mounted on their end-effectors, which reduces human labors dramatically and keeps human operators from being exposed to harmful working environments. Compared to the full automation in the additive manufacturing booth, the path/trajectory planning problem always causes a headache for people in the design room. A human operator has to use trial-and-error approach in order to find a path or trajectory for the tool. It is even harder for an operator to figure out an optimal path or trajectory when some performance measures are considered. For example, in Ford Motor Company's *Aston Martin^{TM}* plant, it takes an experienced engineer about 8 weeks to design a spray gun trajectory for a door panel. Furthermore, the generated path or trajectory is usually operator-dependent, error-prone. It is highly desirable to reduce the time-to-market by replacing human operators with computers in path and trajectory planning. This is a challenging research topic and has been receiving more and more attention from academia and industry.

Some researchers have studied the path planning for spray painting. Asakawa *et al.* [1] presented a teachingless path generation method to paint a car bumper. Antonio *et al.* [2] developed a framework for optimal trajectory planning to deal with the optimal paint thickness problem. Penin *et al.* [3] proposed an automatic path planning method to spray glass fiber on a panel with cement. On the other hand, some research work has been reported in the robotic coverage problem, which is closely related to the tool planning problem in additive manufacturing. Choset [4] presented a method to generate coverage-guaranteed paths for a surface with holes. Mizugaki [5] developed a path planning method for a polishing robot. Takeuchi *et al.* [6] discussed the polishing path generation using the CAD model of a surface.

In summary, most of the previous works use ad hoc methods to obtain paths for specific applications. The major problems with the previous works include:

(1) There lacks a general framework to guide the tool path planning in additive manufacturing.

(2) Only simple surfaces, normally in 2D plane, are considered in previous work. However, many parts in additive manufacturing consist of freeform surfaces and also have complicated topological structures, it is difficult to plan tool paths without partitioning the surfaces into multiple simple patches.

(3) Little research work considers the robot motion performance in tool planning. As a matter of fact, the motion performance is very important to achieve manufacturing efficiency.

We are developing a general framework of the tool planning system, as shown in Figure 1, to address the above problems. There are four major inputs to this automated tool planning system, the part model, tool models, task constraints and optimization criteria. The planning system generates a tool plan to manufacture this part. The generated tool plan, which normally consists of tool paths/trajectories, is validated by a simulation module and a verification module.

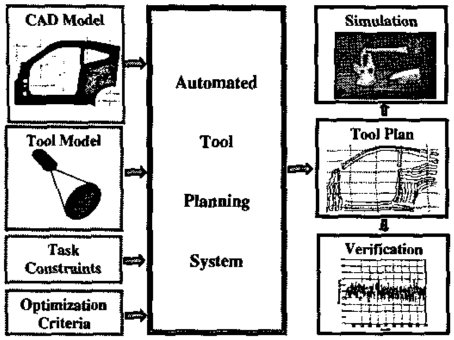


Figure 1: The general framework of the tool planning system.

This tool planning system consists of four basic modules: 1) part preprocessing; 2) geometric understanding; 3) path generation and 4) path integration. The major steps of the tool planning process are shown in Figure 2.

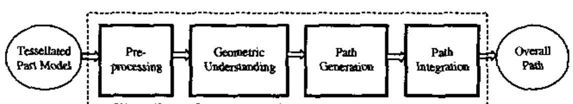


Figure 2: The major steps of the tool planning process.

The part preprocessing module aims at reducing the calculation complexity through applying certain geometric transformation on the original CAD data. The geometric understanding module is the core of the tool planning system and its goal is to (1) capture the general topology of the part geometry; (2) partition the part surface into multiple patches so that each patch is easy to plan tool paths. The path generation module plans the path on each patch. The path integration module deals with how to connect the different paths on each patch to form an efficient overall path.

The objective of this research is to develop a framework to formalize the tool path planning process for complicated free-form parts that are usually seen in additive manufacturing. This paper mainly addresses the geometric understanding problem. The organization of the paper is as follows: Section 2 discusses the part geometry preprocessing; Section 3 presents the decomposition-based surface partitioning approach; Implementation and experimental tests are reported in Section 4 and Section 5 gives the conclusions.

2 Part Geometry Preprocessing

The CAD data of a part usually consists of multiple parametric surfaces which are normally in the form of analytical equations. As a result, parametric surfaces are not computation-friendly. A tessellation representation method is adopted, which models the part surfaces using polygonal facets, usually triangles.

Due to the complexity of the part geometry, it is hard to directly work on the free-form surfaces. It is observed that most of the parts in additive manufacturing consist of low-curvature free-form surfaces. This observation implies that, in order to understand the geometric structure of a free-form surface, it is sufficient to analyze its 2D projection, which can be obtained by projecting the free-form surfaces on a plane. To determine the normal of the projection plane so that the 2D projection captures the topological characteristic of the original part, we introduce the concept of weighted average normal, which can be characterized as $\vec{n}_a = (\sum_{i=1}^{k} A_i \vec{n}_i)/(\sum_{i=1}^{k} A_i)$ [7]. Here n_a is the weighted average normal. k is the number of triangles. $\vec{n}_i$ is the normal of triangle i and A_i is its area. The normal of the projection plane is chosen as the weighted average normal direction. It has been proved in our previous work that the projection on this direction achieves the maximum projection area [7].

Once the normal of the projection plane is determined, all the points on the tessellated free-form part surfaces are orthographically projected onto the projection plane. The outer contour and all the inner contours are identified, which are polygons. The number of edges depends on the resolution of the triangulation of the original part surfaces. To maintain a moderate problem complexity, we reduce the number of edges using a rough polygonal approximation process, which first detects the critical points on the contours and then connects the critical points sequentially to get the approximated contours. This approximation process preserves the major topological characteristic of the contours. To find the critical points on the contours, the parallel algorithm developed in [8] is adopted.

3 Geometric understanding

This section discusses how to partition complicated part surfaces into several patches so that each patch is well-behaved in the sense of the easiness of robot movement.

3.1 Polygon Partitioning

Projection converts the surface partitioning problem in a 2D plane problem. To guide the partitioning of a 2D polygon which may have holes (as shown in Figure 3) into subpolygons, the definition of a "good" partition should be given, which means certain performance measures related to the manufacturing process should be introduced. Based on these measures different partitions can be evaluated and compared. The following are good candidates for the performance measures.

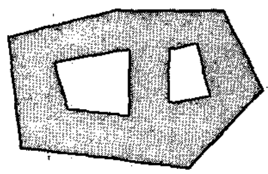


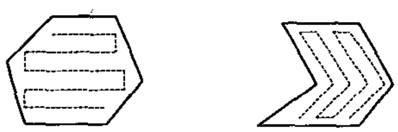
Figure 3: A polygon with two holes

(1) Regularity of subpolygon. If the subpolygon is a rectangle-like polygon, it is easier to control the tool to move along the paths. Sharp angles always bring difficulties in automated tool planning and motion control due to the small area at the corner [9]. For example, the subpolygon in Figure 4(a) is better than the subpolygon in Figure 4(b) in the sense of robot motion performance. Essentially, the regularity requirement implies a favor on right or obtuse interior angles.

(a) (b)

Figure 4: (a) has higher regularity than (b).

(2) Convexity. It is usually easier to plan paths and control the robot on convex subpolygons than on concave ones since concavity implies more frequent changing of the tool moving direction. For example, the subpolygon in Figure 5(a) requires more tool direction changes than the subpolygon in Figure 5(b). Therefore it is necessary to avoid those interior angles that are greater than π in surface partitioning.



(a) (b)

Figure 5: (a) is convex and (b) is concave.

(3) The number of turns a tool has to make to finish the subpolygon. Less turns usually imply less travelling time for the tool since making turns requires a tool to slow down and speed up again [10].

(4) The length of inner borders between subpolygons. The inner border is where two subpolygons meet. It is usually hard to achieve thickness uniformity along the inner borders which receive material from paths on two different patches [11]. Hence it is desirable to reduce the total length of inner borders.

(5) The number of subpolygons. Less subpolygons mean less transitions between patches for the tool and thus more efficiency.

To mathematically characterize the regularity and convexity of a subpolygon, we introduce the following regularity and convexity measure: $RC = (\sum_{i=1}^{p} \lambda(\theta_i))/p$. Here p is the number of vertex on the subpolygon and θ_i ($i = 1, 2, ..., p$) are the interior angles of the subpolygon. This measure is the arithmetic average of the penalty function $\lambda(\cdot)$ of all the interior angles, which is defined as follows:

$$\lambda(\theta_i) = \begin{cases} 1 - \frac{2}{\pi}\theta_i & 0 \leq \theta_i \leq \frac{\pi}{2} \\ 0 & \frac{\pi}{2} < \theta_i \leq \pi \\ \frac{\theta_i}{\pi} - 1 & \pi < \theta_i \leq 2\pi \end{cases} \tag{3.1}$$

$\lambda(\cdot)$ can be visualized as in Figure 6. Clearly, this piece-wise penalty function favors the interior angles between $\pi/2$ and π. Beyond this range, sharp interior angles or concavity will appear.

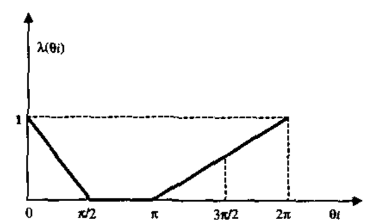


Figure 6: The penalty function $\lambda(\cdot)$.

The number of turns a tool has to make to finish a subpolygon can be represented by the minimum altitude, ALT_{min}, of the subpolygon [10]. Figure 7 illustrates the relationship between the path sweep-direction and the number of turns.

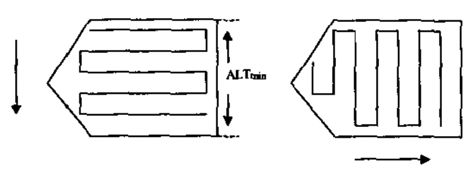


Figure 7: The minimum altitude represents the least number of turns.

To combine the above measures, a generalized performance measure M_g is defined for each subpolygon SP_i,

$$M_g = \omega_1 h_1(RC) + \omega_2 h_2(ALT_{min}) + \omega_3 h_3(L_{ib}) + \omega_4 \cdot 1. \tag{3.2}$$

Here ω_i $(i = 1, 2, 3, 4)$ are weights and the $h_i(\cdot)$ $(i = 1, 2, 3)$ functions are defined such that when the performance measure achieves its optimality, $h_i(\cdot)$ achieves its minimum. $\omega_4 \cdot 1$ reflects the contribution to the total number of the partitioned subpolygons. Based on the generalized performance measure, it is easy to emphasize specific performance measures by adjusting the corresponding weights. A good partition of the polygon implies that the generalized performance measure approaches minimum. Figure 8 gives two different partitions for the same example polygon. It is clear that as long as the convexity and number of subpolygons are concerned, partition A is better than partition B.

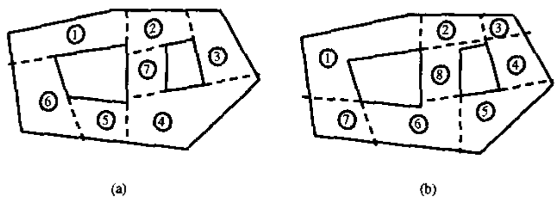


Figure 8: The two different partitions

Based on the above discussions, the partitioning problem can be formulated as the following geometric problem.

Given an n-edge polygon with k polygonal holes, each hole has n_i edges $(i = 1, 2, ..., k)$, partition the polygon into m subpolygons so that the overall generalized performance measure $M_g = \sum_{i=1}^{m} M_{gi}$ is minimized.

There has been some research effort in polygon partitioning. Most of them address it from a traditional mathematical point of view [12, 13, 14]. Another portion of the research is motivated by practical applications, such as robotic coverage problems [15, 16, 17]. Recently, Huang [10] proposed coverage algorithms based on optimal line-sweep decompositions. His work is inspired by Choset and Pignon [15] as well as Hert et al.'s [16] coverage work. Using dynamic programming, Huang provides algorithms to partition arbitrary polygons (with holes) into multiple areas and minimizes the total altitude to cover all the areas, which implies the least number of turns for the tool. In summary, the polygon partitioning approaches developed in the previous work can not handle the surface parititioning problem we face due to the following reasons:

(1) Most of the previous work partitions the polygon into fixed number of subpolygons while in our problem, there is no fixed, or a priori-known number of subpolygons.

(2) Most of the previous work only considers the number of components or the total length of cut length. For an additive manufacturing application, several performance measures should be considered, such as the regularity, convexity of the patches, the total inner border length, the total number of turns, etc. It is desirable to have a general solution to partition the polygon when multiple measures are used.

3.2 The Decomposition-based Approach

Our approach first decomposes a polygon into multiple cells. Secondly, the cells are combined to form different subpolygons. For each subpolygon, a weight is assigned to it, which indicates the generalized performance measure of the subpolygon. Then a weighted set partitioning problem can be formulated and solved to select a minimum-weight set of the subpolygons which partitions the polygon.

Theoretically, there are infinite ways to decompose a polygon into different polygonal cells. To maintain a limited complexity of the problem, we partition the polygon by extending the line of each edge of the inner polygons. For the polygon in Figure 2, a decomposition is shown in Figure 9(a).

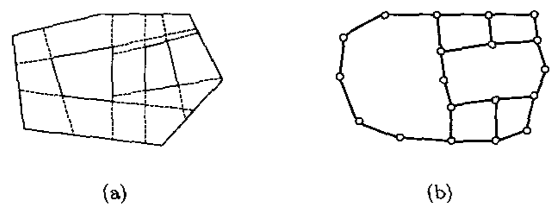


Figure 9: (a) Cell-decomposition of a polygon and its CAG.

Once the polygon is partitioned into cells, a new mathematical tool, the Cell Adjacency Graph, is introduced which represents the connectivity among the cells.

Definition 3.1 *A* **Cell Adjacency Graph (CAG)** *is a unidirected graph $G = (V, E)$ where V is the set of nodes and $E \subset V \times V$ is the set of edges. Each node V_i represents a cell in the polygon. Between V_i and V_j exists edge e_{ij} if the two nodes are adjacent.*

The CAG for the example decomposition in Figure 9(a) is shown in Figure 9(b). Obviously, the CAG is a planar, connected graph. The size of the CAG can be roughly estimated by modelling the cell decomposition as 2D plane partitioning using l lines, here $l = n1 + n2 + ... + n_k$ is the total number of edges on inner holes, a upper bound estimation on the total number of cell is given as $\overline{N_c} = (\sum_{i=1}^{k} n_i)(\sum_{i=1}^{k} n_i + 1)/2 + 1 - k$ [18].

To generate different subpolygons, it is convenient to do it on the CAG. Each subpolygon is, essentially, a connected subgraph of the CAG. Therefore, to find all different subpolygons with k cells is equivalent to the following problem:

given a connected planar graph $G = (V, E)$, find all the different connected subgraph $G_i = (V_i, E_i)$ on G *with* $|V_i| = k$, $(1 \leq k \leq |V|)$. *Here* $|V|$ *is the number of element in set* V.

Some existing algorithms can be used to find all the connected subgraphs with k vertices [19]. Denote the total number of k-cell subgraphs as N_k then the total number of subpolygons $N_{sp} = \sum_{i=1}^{|V|} N_k$.

In order to find a partition of the polygon that has the minimum overall generalized performance measure, we model the problem as a weighted set partitioning problem [20], which is a special structure in integer programming and can model many industrial scheduling/planning problems like bus crew scheduling, air crew scheduling, facility location etc. The classical set partitioning problem is stated as follows,

given 1) a finite set M with m members; 2) a constraint set defining a family F of n 'acceptable' subsets of M; find a minimum collection of members of F which is a partitioning of M [20].

Assume that we have identified all the subpolygons SP_i $(i = 1, 2...N_{sp})$. Each subpolygon SP_i has an index set of the cells $I_i = \{I_{i_1}, I_{i_2}, ...I_{i_j}\}$, $I_{i_j} \in [1, 2, ..., N_c]$, here N_c is the total number of cells. Each subpolygon SP_i is associated with a weight C_i which is equal to the generalized performance measure M_{gi}. The weighted set partitioning problem can be formulated as follows,

$$\text{min} \quad z = \sum_{j=1}^{N_{sp}} c_j x_j \tag{3.3}$$

$$\text{subject to:} \quad \sum_{j=1}^{N_{sp}} a_{ij} x_j = 1, i = 1, 2, ..., N_c$$

$$\text{with} \quad x_j = 0 \text{ or } 1, \; C_j = M_{gj}, \; j = 1, 2, ..., N_{sp}.$$

Here c_j is the cost of subpolygon SP_j, which is set to be the generalized performance measure M_{gj} of subpolygon SP_j. $a_{ij} = 1$ means cell i belongs to subpolygon SP_j, otherwise not. Variable $x_j = 1$ means the subpolygon SP_j is selected in the final partitioning, otherwise not. The weighted set partitioning problem is NP-hard [20]. There exist many algorithms, as well as softwares that solve the weighted set partitioning problem. The solution to this algebraic problem implies a partitioning of the polygon with a minimum overall generalized performance measure.

4 Implementation and Results

The tool planning system is implemented and the algorithms for 2D projection, rough polygonal approximation, polygon partitioning and 2D-3D mapping are implemented in C++ on a PC with *Pentium*^{TM} III 500MHZ processor. The weighted set partitioning problem is solved by a commercial software, *IPLOG Studio*^{TM} [21] which provides fast, optimal or near-optimal solutions. We test our surface partitioning approach on many parts from Ford Motor Company. One of the parts tested is a car door panel, which is shown in Figure 10.

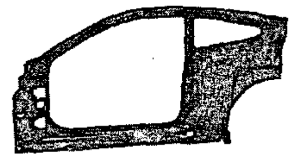
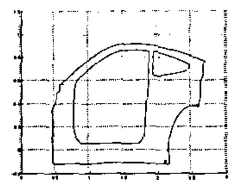
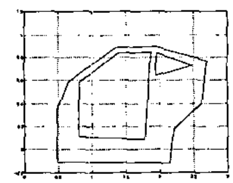


Figure 10: A door panel.

The 2D projection of the part along its weighted average normal direction is shown in Figure 11(a) and the rough polygonal approximation is shown in Figure 11(b). Here some details are ignored by filtering out small holes on the left side of the door panel.



(a)



(b)

Figure 11: (a) The 2D projection (b) The approximated polygon.

We tested the partitioning approach using different weights ω_i. It is worth noting that the $h_i(\cdot)$ functions are normalized by multiplying scale factors.

(1) Case 1: $\omega_1 = 1, \omega_2 = 0, \omega_3 = 0, \omega_4 = 0$. In this case, only the regularity and the convexity are considered in the partitioning. The generated polygon partition is shown in Figure 12. As can be seen from this figure, the partition favors obtuse angles and reduces the concavity of the subpolygons to its minimum.

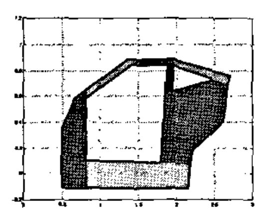


Figure 12: The polygon partition for case 1.

(2) Case 2: $\omega_1 = 0, \omega_2 = 1, \omega_3 = 0, \omega_4 = 0$. In this case, only the number of turns, or the total minimum altitude is considered. The generated polygon partition is shown in Figure 13. From this figure, it is obvious that the partition favors small altitude of subpolygons, although the length of inner borders are sacrificed.

(3) Case 3: $\omega_1 = 0.25, \omega_2 = 0.25, \omega_3 = 0.25, \omega_4 = 0.25$. In this case, all the performance measures are considered and they have equal weights. The generated polygon partition is shown in Figure 14.

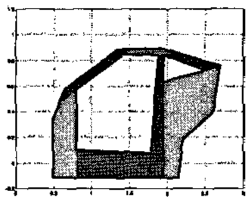


Figure 13: The polygon partition for case 2.

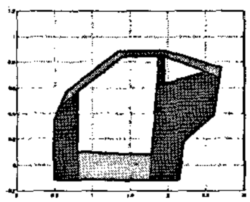


Figure 14: The polygon partition for case 3.

The partition in this case achieves a tradeoff among all the performance measures. It is clear that this partition reaches an overall performance optimality. The partition of the door panel for case 3 is shown in Figure 15.

Figure 15: The partitioned door panel.

5 Conclusions

In this paper, a general framework is proposed for tool planning in additive manufacturing applications. To handle the complicated geometric structure of the part surfaces, a divide-and-conquer strategy is developed, which can be modelled as a surface partitioning problem regarding certain robot motion performances. A decomposition-based approach is developed, which renders the geometric partitioning problem as an algebraic integer programming problem. Experimental results demonstrate the effectiveness of the proposed approach.

References

[1] N. Asakawa and Y. Takeuchi. Teachingless spray-painting of sculptured surface by an industrial robot. In *IEEE International Conference on Robotics and Automation*, pages 1875 -1879, Albuquerque, New Mexico, April 1997.

[2] J. K. Antonio, R. Ramabhadran, and T. L. Ling. A framework for optimal trajectory planning for automated spray coating. *International Journal of Robotics and Automation*, 12(4):124–134, 1997.

[3] L. F. Penin, C. Balaguer, J. M. Pastor, F. J. Rodriguez, A. Barrientos, and R. Aracil. Robotized spraying of prefabricated panels. *IEEE Robotics and Automation Magazine*, 5(3):18 -29, 1998.

[4] H. Choset. Coverage of known spaces: The boustrophedon cellular decomposition. *Autonomos Robots*, 9:247–253, 2000.

[5] Y. Mizugaki, M. Sakamoto, and K. Kamijo. Fractal path application in a metal mold polishing robot system. In *Fifth International Conference on Advanced Robotics*, pages 431 – 436, Pisa, Italy, June 1991.

[6] Y. Takeuchi, D. Ge, and N. Asakawa. Automated polishing process with a human-like dexterous robot. In *IEEE*, pages 950–956, Barcelona, Spain, Oct. 1993.

[7] W.Sheng, N.Xi, M.Song, Y.Chen, and J.S. Rankin III. Automated cad-guided automobile part dimensional inspection. In *Proceedings of 2000 international conference on Robotics and Automation*, volume 2, pages 1157–1162, 2000.

[8] C.H.The and R.T.Chin. On the detection of dominant points on digital curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8), 1989.

[9] Y.Yang, H.T.Loh, F.Y.H.Fuh, and Y.G.Wang. Equidistant path generation for improving scanning efficiency in layered manufacturing. *Rapid Prototyping Journal*, 8(1):30–37, 2002.

[10] W.H.Huang. Optimal line-sweep-based decomposition for coverage algorithms. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, 2001.

[11] H. Chen, N. Xi, Z. Wei, Y. Chen, and J. Dahl. Trajectory integration for a surface with multiple patches. In *Proceeding of the IEEE International Conference on Robotics and Automation*, 2003.

[12] C.Levcopoulos and A.Lingas. Bounds on the length of convex partitions of polygons. In *Proceedings of the 4th Conference on Found. Softw. Tech. Theoret.Comput.Sci., Lecture Notes in Computer Science*, volume 181, pages 279–295. Springer-Verlag, 1984.

[13] V.Soltan and A.Gorpinevich. Minimum dissection of rectilinear polygon with arbitrary holes into rectangles. *Discrete Computational Geometry*, (9):57–79, 1993.

[14] B.Chazelle and D.Dobkin. Decomposing a polygon into its convex parts. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing*, pages 38–48, 1979.

[15] H.Choset and P.Pignon. Coverage path planning: the boustrophedon cellular decomposition. In *Proceedings of the 1997 International Conference on Field and Service Robotics*, 1997.

[16] S.Hert, S.Tiwari, and V.Lumelsky. A terrain-covering algorithm for an auv. *Autonomous Robots*, 3:91–119, 1996.

[17] H.Bast and S.Hert. The area partitioning problem. In *Proceedings of the 12th Canadian Conference on Computational Geometry*, pages 163–172, 2000.

[18] 2002. Personal communication with S.Zhu.

[19] 2002. Personal communication with D.Israel.

[20] E.Balas. *Combinatorial Optimization*, chapter 7, pages 151–210. 1979.

[21] P.V.Hentenryck. *The OPL Optimization Programming Language*. MIT Press, 1999.