Polygon Reconstruction from Line Cross-Sections *

Avishay Sidlesky † Gill Barequet † Craig Gotsman †

Abstract

We study the following geometric probing problem: Reconstruct a planar polygon from its intersections with a collection of arbitrarily-oriented "cutting" lines. We propose an algorithm which enumerates *all* possible reconstructions that are consistent with the input and comply with a realistic sampling condition. We analyze the complexity of the algorithm and provide some experimental results.

1 Introduction

In the field of medical imaging, numerous applications can benefit from a three-dimensional model of human organs. Common imaging techniques, such as MRI and CT, output a dense set of parallel slices of the region of interest and as a consequence, practically all prior art addresses the problem of reconstructing a three-dimensional triangular mesh from parallel planar slices of an object (e.g., [2, 3, 4, 5]).

In recent years, sensors capable of accurately measuring position and orientation (referred to as P&O) have been developed. These sensors, when mounted on relatively cheap hand-held devices, such as ultrasound transducers, provide valuable P&O information of the cross-section planes, and lead to the reconstruction of a three-dimensional triangular mesh from arbitrary, nonparallel, slices (see, e.g., [6] for a volumetric-compounding approach).

An interesting, yet unexplored, problem by itself is the two-dimensional version of the problem, namely, two-dimensional polygon reconstruction from line cross-sections. (A precise definition is given in Section 2.) A sampling condition, that guarantees sufficient sampling of the polygon edges, must be formulated, otherwise an infinite number of solutions may exist. In addition, a merit function may be defined to allow comparison between solutions, if several exist, to choose the "best" one.

To the best of our knowledge, there exists only a single work by Coll and Sellarès [1] on a problem that closely resembles ours. In that paper, the authors assume that the cutting lines are sufficiently "dense" and uniformly distributed over the unknown shape. They introduce an algorithm that processes the input cutting lines sequentially, and is incremental in the sense that for each new line it updates the reconstruction in expected time which is logarithmic in the total number of cutting lines. At each stage, their algorithm outputs a single triangulation from which a polygonal reconstruction of the unknown shape could be extracted. In contrast, our aim is to treat also cases in which a *small* number of cross-sections are given, and to enumerate all possible solutions, ordered according to some measure of quality.

*This work was supported by the EU Network of Excellence AIM@SHAPE - IST NoE 506766.

†Department of Computer Science, Technion—Israel Institute of Technology, {sid,barequet,gotsman}@cs.technion.ac.il

2 The Reconstruction Algorithm

The problem of two-dimensional polygon reconstruction from line cross-sections is defined as follows:
Given a set $\mathcal{L}$ of cutting lines and the intersection-segments, $\mathcal{S}$, of an unknown polygon (or possibly several disjoint or nested polygons) $\mathcal{P}$ with $\mathcal{L}$, find all reconstructions $\mathcal{R}$ (or the best one) that are consistent with the input, i.e., $\mathcal{R} \bigcap \mathcal{L} = \mathcal{P} \bigcap \mathcal{L} = \mathcal{S}$. Note that it is possible for a cutting line not to intersect $\mathcal{P}$ at all, and in this case the line does not contribute to $\mathcal{S}$.

2.1 The Sampling Condition

In order to limit the number of solutions to the polygon-reconstruction problem, we impose a natural *sampling condition*. This condition requires that each edge of the reconstructed polygon(s) $\mathcal{R}$ be intersected by at least two cutting lines in distinct locations. The sampling condition does not guarantee a unique solution, but it prevents the creation of infinitely-many solutions, differing slightly from each other.
For instance, if some edges of a reconstructed polygon are not intersected by any cutting line, that region of the polygon can be arbitrarily reconstructed. If an edge is intersected by a single cutting line, the intersection point becomes a pivot around which the reconstructed edge may revolve, generating an infinite number of reconstructions.

2.2 Definitions

To describe the algorithm, we need to distinguish between various regions of the plane created by the input cutting lines and intersection segments. Refer to Fig. 6 for an illustration (intersection segments emphasized).

• *Cell*: A (convex) region in the plane bounded by portions of some cutting lines.

• *Signature segment*: A portion of an intersection segment that lies on an edge of a cell. Note that an edge of a cell may contain several (disjoint) signature segments or none at all.

• *Signature chain*: The concatenation of several signature segments joined at their endpoints and occupying multiple cell edges.

• *Boundary cell*: A cell that contains at least one signature chain on its boundary. A cell with a single (open) signature chain is called a *regular cell*.

• *Bridge cell*: A boundary cell with more than one signature chain. It is called this because it has the potential of connecting between two or more disconnected components of the reconstructed polygon.

• *Cell configuration*: A pairing of all endpoints of signature chains of a cell that can be realized by noncrossing chords. The cell configuration determines the local topology of the reconstruction. Figure 1 illustrates all possible configurations of a bridge cell having three signature chains.

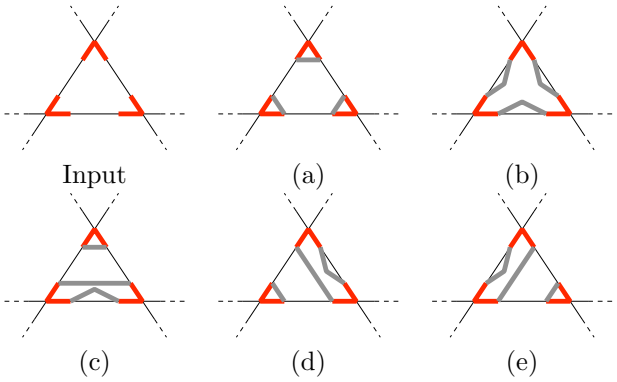


Figure 1: Configurations of a bridge with 3 sig. chains

• *Cell reconstruction*: A pairwise connection of all the endpoints of the signature chains of a cell, for a specific cell configuration. Each connector is either a straight line segment, which eventually becomes a portion of an edge of the reconstructed polygon, or two consecutive line segments sharing an intra-cell vertex, which corresponds to a corner of the reconstructed polygon. The same cell configuration can be realized in more than one cell reconstruction.

• *Polygon reconstruction*: A reconstruction of all boundary cells for a specific combination of cell configurations.

2.3 Overview of the Algorithm

Our algorithm operates only on the boundary cells. It reconstructs the polygon by creating portions of the polygon edges within each such cell. We then use inter-cell relations to find all possible polygon reconstructions consistent with the input.

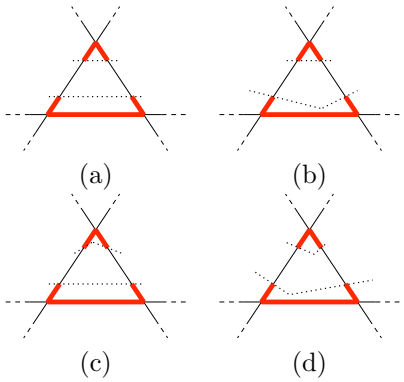


Figure 2: Reconstructions of a bridge with 2 sig. chains

The algorithm consists of the following steps:

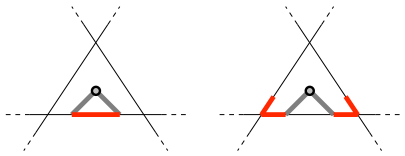
2.3.1 Data structure initialization

In the initialization step, we create a DCEL structure containing the cells, their adjacency relations, and the signature chains along their boundaries. Denote by ℓ the number of cutting lines in $\mathcal{L}$, and by s the number of intersection segments in $\mathcal{S}$. During this step, we introduce $O(\ell^2)$ new endpoints of signature segments by splitting the intersection segments at the intersection of the cutting lines. Thus, the total space complexity of the data structure, as well as the time it takes to initialize it, is $O(\ell^2 + s)$. This is also the space complexity of the entire algorithm since it does not require any additional memory for the remaining steps.

2.3.2 Eliminating bridge cell configurations

Let m_i be the number of signature chains in the ith bridge cell. Observe that each bridge cell configuration connects the endpoints of the signature chains by noncrossing chords. The number of configurations is, therefore, the m_ith Catalan number, C_{m_i}. In each such configuration we tag the connectors of the endpoints as having a compulsory intra-cell vertex if one of the two following conditions holds: (a) a signature segment is connected to itself at its ends; or (b) two collinear signature segments are connected. See Figure 3(a,b), respectively.

Next, we check all the cells having a single configuration, and delete the configurations of the adjacent bridge cell if both connectors at the same signature were tagged as having a compulsory vertex (see Figure 4). These configurations do not satisfy the sampling condition since they imply that the reconstructed edge is sampled by a single cutting line.



(a) (b)

Figure 3: Cases of a compulsory vertex

(a) A valid configuration (b) An invalid configuration

Figure 4: Eliminating bridge configurations

2.3.3 Identifying polygon reconstructions

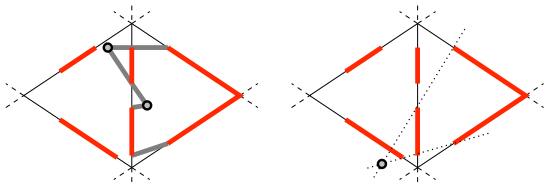
At this point we are left with a number of potential polygon reconstructions that is equal to the product of the number of remaining configurations of all the cells. Each potential reconstruction stems from a distinct combination of configurations of cells.

By traversing all the cells, we attempt to connect the endpoints of signature chains according to the cell configuration. If a connector was previously tagged as having a compulsory vertex, and if either

1. one of the adjacent cell connectors of the same signature endpoint was also tagged as such; or

2. an intra-cell vertex (created by intersecting the straight line connectors from the adjacent cells) is found outside the cell,

then we conclude that the entire polygon reconstruction is invalid (see Figure 5). If all the cells have been reconstructed properly, this combination yields a valid solution.



(1) (2)

Figure 5: Invalid combinations

3 Complexity Analysis

The time complexity of the entire algorithm is dominated by the procedure described in Section 2.3.3. For the analysis, we use the well known approximation of C_n (for $n \gg 1$)

$$C_n \equiv \frac{1}{(n+1)}\binom{2n}{n} \sim \frac{4^n}{\sqrt{\pi}n^{3/2}}.$$

Let k be the number of boundary cells in the arrangement of cutting lines, and let m_i be the number of signature chains along the boundary of the ith cell, where $\sum_{i=1}^{k} m_i = M$. We may consider M a "budget" of signature chains that is distributed among the boundary cells. Notice that $M = 2s$ since each endpoint of an intersection segment is considered twice. Multiplying the above expression for all k cells gives the following upper bound on the number of reconstructions, R, that are consistent with the input:

$$R(M, m_i) = \frac{4^M}{\prod_{i=1}^{k}(\sqrt{\pi}m_i^{3/2})}. \quad (1)$$

We now observe the values of k and m_i's that maximize the above term.

Theorem 1 *The maximal value of the term (1) is obtained by minimizing the number of cells and by partitioning M such that all but a constant number of the boundary cells have a single signature chain.*

Theorem 1 implies that in the worst case, there is a small (constant) number of cells, and that a single cell has $O(M)$ signature chains. Substituting this into Equation (1), we obtain

$$R(M) = O(\frac{4^M}{M^{3/2}}). \quad (2)$$

To conclude the complexity analysis, we take into account that $O(M \log M)$ time is needed for reconstructing that large cell (to ensure simple, non-self-intersecting, reconstructions), and obtain the upper bound on the total run time of the algorithm:

$$T(M) = O(\frac{4^M \log M}{\sqrt{M}}).$$

In the full version of the paper we provide an example which establishes a lower bound on $R(M)$ which almost matches the upper bound (2).

4 Example

Figure 6 shows a typical input with the regular cells and bridge cells highlighted. In this example, where the input consists of $\ell = 8$ cutting lines and $s = 14$ intersection segments, there are initially 800 potential reconstructions. By eliminating bridge-cell configurations that violate the sampling condition, our algorithm reduced the number of potential reconstructions to 288. At the end, we are left with only four simple reconstructions that are consistent with the input, depicted in Figure 7.

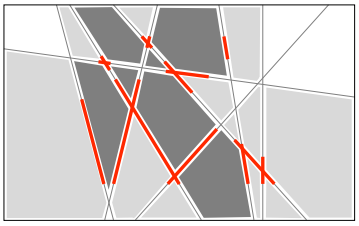
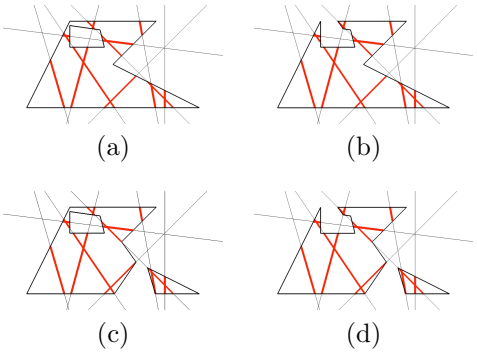


Figure 6: An example input with regular cells (light gray) and bridge cells (dark gray) highlighted



(a) (b) (c) (d)

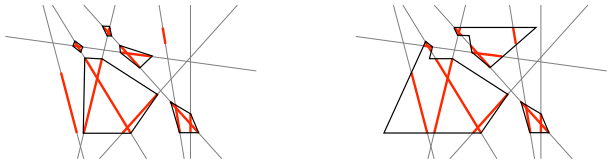
Figure 7: Four reconstructions from the input of Fig. 6 (without trimmed corners)

5 Comparison with Previous Work

As mentioned in the introduction, Coll and Sellarès [1] dealt with a problem that closely resembles ours. Their algorithm outputs, in $O(\ell \log \ell + s)$ expected time, a *single* polygonal approximation, whose vertices are the endpoints of the intersection segments of the unknown shape.

Our algorithm, on the other hand, outputs all possible reconstructions that are consistent with the input and forces the vertices of the reconstructed polygon to be the endpoints of the intersection segments only if solutions with trimmed corners are sought.

Figures 8(a,b), respectively, shows the reconstructions obtained by the algorithm of Coll and Sellarès and by our algorithm (with trimmed corners) that resembles theirs the most (rather subjectively), when applied to the input shown in Figure 6.



(a) Coll's reconstruction (b) One of our reconstructions

Figure 8: The reconstruction of [1] vs. ours

Note that due to the nature of their algorithm, single isolated intersection segments do not contribute to the reconstruction. Furthermore, since their algorithm is incapable of bridging gaps between groups of connected intersection segments, the number of connected components in the output reconstruction is maximized.

We may modify our algorithm to ignore all signature chains that consist of a single signature segment, and restrict the bridge cells to a configuration in which each signature chain is connected to itself. Thus, we force a single reconstruction that is composed only of vertices that are the endpoints of the intersection segments, which is identical to the reconstruction of [1].

The run time of this degenerate version of our algorithm is dominated by the time needed to initialize the data structure ($O(\ell^2 + s)$ in the worst case), which is slightly worse, but still comparable to that of the algorithm of [1], which is $O(\ell \log \ell + s)$ expected time.

6 Conclusions

In this paper we investigate the problem of two-dimensional polygon reconstruction from line cross-sections. We focus on, but do not limit ourselves to, cases where the number of cutting lines is small, and seek all reconstructions that are consistent with the input and comply with a natural sampling condition. We describe an algorithm for providing these reconstructions in detail.

The complexity analysis shows that the number of reconstructions, and hence, the run time of the algorithm, is exponential in nature. Note that in some cases, there may be several valid reconstructions for a single combination of configurations of cells. In this case, our algorithm outputs the unique solution having trimmed corners for this combination. If we restrict the algorithm to provide a single reconstruction, identical to that of [1], the run time is quadratic in the size of the input. The related question of conditions for uniqueness of the solution remains open.

References

[1] N. Coll and J.A. Sellarès, Planar shape reconstruction from random sections, *17th European Workshop on Computational Geometry*, Berlin, Germany, 121–124, March 2001.

[2] G. Barequet and M. Sharir, Piecewise-linear interpolation between polygonal slices, *Computer Vision and Image Understanding*, 63 (2), 251–272, March 1996.

[3] H. Müller and A. Klingert, Surface interpolation from cross sections, in: *Focus on Scientific Visualization* (H. Hagen, H. Müller, and G.M. Nielson, eds.), Springer Verlag, 139–189, 1993.

[4] C.L. Bajaj, E.J. Coyle, and K.N. Lin, Arbitrary topology shape reconstruction from planar cross sections, *Graphical Models and Image Processing*, 58 (6), 524–543, November 1996.

[5] G. Barequet, M.T. Goodrich, A. Levi-Steiner, and D. Steiner, Contour interpolation by straight skeletons, *Graphical Models*, 66 (4), 245–260, July 2004.

[6] R.N. Rohling, A.H. Gee, and L. Berman, 3-D spatial compounding of ultrasound images, *Medical Image Analysis*, 1 (3), 177–193, April 1997.