Reconstruction of Three-Dimensional Objects from Parallel Planar Sections – Comparative Analysis of Solutions of the Correspondence Problem

Cătălin Tudose, Tudor Metea

ITC Networks,
Calea Floreasca 167, Bucharest, Romania,
Email: catalin_tudose@yahoo.com

Department of Automatic Control and Systems Engineering,
"POLITEHNICA" University of Bucharest,
Spl. Independentei 313, 77206 Bucharest, Romania,
Email: mtudor@mailbox.ro

Abstract: This paper presents some aspects concerning the correspondence problem and a comparative analysis of some different solutions to it, as part of the spatial reconstruction of three-dimensional objects represented by parallel planar sections. The analysis concerns comparative performance results of the implementation of two different algorithms on practical cases, taken from geography.

Keywords: Reconstruction technique, serial section, correspondence problem, graph-based algorithm, surfaces-based algorithm.

1 Introduction

Serial section reconstruction is widely used for visualizing complex three-dimensional objects. A three-dimensional object is commonly sampled by a set of parallel planar sections, in which closed contour loops define the intersection between these sections and the object. The main reason for reconstruction of serial sections is to enable computer-based three-dimensional visualization that includes views of the internal features. The results of the work are useful in geo-scientific fields, as geography, geology, paleontology and sedimentlogy.

Firstly, the main resulted sub-problems are reviewed. Secondly, a detailed presentation of two different algorithms for the correspondence problem, and finally, some comparative tests are performed and analyzed.

2 Review of Serial Section Reconstruction Techniques

The process of serial section reconstruction can be broken down into a number of sub-problems, as shown in Fig. 1:

- the *correspondence problem* is concerned with specifying the connectivity between contours in adjacent sections. Given two contours, we must determine whether they should be connected by surface or not. A solution to this sub-problem can be used to direct the other parts of the reconstruction process.
- the *tiling problem* consists of finding the best set of triangulated facets to define the surface between a pair of connected contours in adjacent sections.
- the *multiple branching problem* occurs when the numbers of contours on a pair of adjacent sections are not equal. This causes problems for standard tiling algorithms. One solution is to decompose the problem into a one-to-one connection so that the tiling algorithm can be used. The surface then needs to be completed with further triangulation.

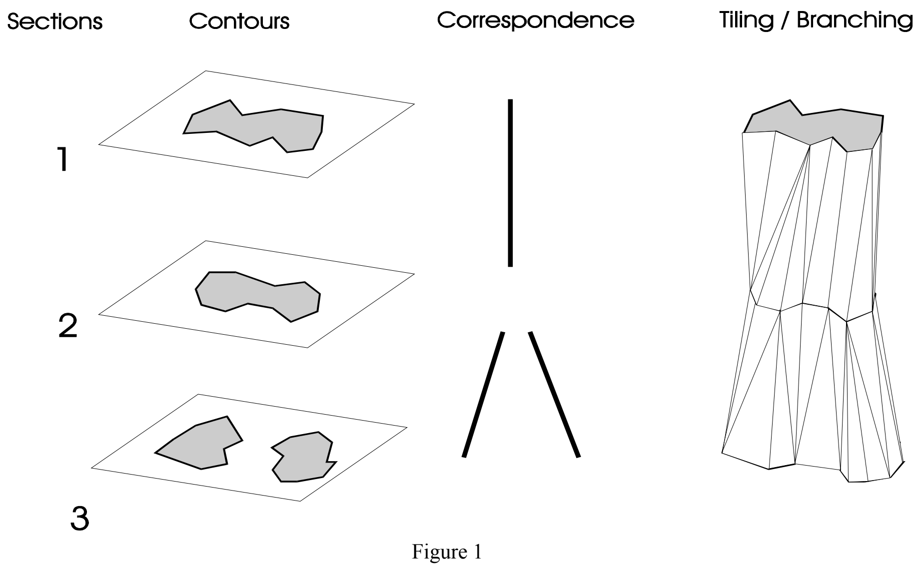

Figure 1

Serial section reconstruction sub-problems

3 Correspondence Solutions

A number of graph-based methods have been used to identify the correspondence between adjacent contours. These methods have some limitations that restrict their use. Most algorithms make connectivity decisions based upon the information provided by just a single pair of sections.

Other algorithms use the superposition of contours. The superposition percentage value of two contours could be evaluated by computing the intersection area between them (Choi and Park, 1994). But it could be used an approximation of this area that requires less computation and is sufficient in most applications, although it produces an unnatural result for some examples (Meyers *et al.,* 1992). For this approximation, it is defined two rectangles encompassing the contours whose sides are parallel to the *x* and *y-axis*. The superposition degree of the two contours is defined as the intersection area of the two contours.

3.1 Graph-Based Solutions

The solution described below consists in three stages:

1. *Preprocessing.* A candidate graph is set up and edges are removed from it if they are regarded as invalid.
2. *Processing.* A minimum spanning tree (MST) algorithm to provide a set of suitable edges for the correspondence solution reduces the graph.
3. *Postprocessing.* Edges are removed from the MST to divide the graph into the individual components that make up the object.

The advantage of this approach is the use of additional spatial information to derive a solution. The correspondence solution uses three forms of spatial information:

1. *Relative position.* The location of the pair of contours on adjacent sections, *i* and *j,* given by the contour centroid, (x_i, y_i) and (x_j, y_j).
2. *Shape.* The major and minor axes provide this information for each of the contours A_i and B_i, A_j and B_j.
3. *Topological relations.* The relationship between a contour and the other contours on the same section is used to validate the existence of possible surfaces between contours on neighboring sections. There are three types of relation, *Inside, Contains* and *Disjoint that* can exist between a pair of contours on the same section. *Inside* and *Contains* are simply the converse of each other. For our application oriented to the reconstruction of the relief, we do not use this kind of relations, as they do not exist in this case (we cannot have such relations between contours representing the relief).

3.1.1 Preprocessing

The first step is to build a candidate undirected graph G = (V, E), where V is the set of contours and E is the set of possible surfaces between contours in adjacent sections. For each edge *e* in set *E,* a weight is determined by the following metric:

$$e(i,j) = (x_i - x_j)^2 + (y_i - y_j)^2 + (A_i - A_j)^2 + (B_i - B_j)^2 \quad (1)$$

where (x, y, A, B) are the centroid coordinates and the length of the major and minor axes for the pair of contours (i,j) in adjacent section. The metric used combines a comparison of both the position and shape information for each pair of neighboring contours. For some particular cases, as those concerning the reconstruction of the relief, we tried to modify the value of the weight of this edge, considering that the calculus can be simplified, thinking that we can use the following formula for the weight of the edge:

$$e(i,j) = (x_i - x_j)^2 + (y_i - y_j)^2 \quad (2)$$

For these cases of reconstruction, this metric gives good results, as we can notice that this kind of contours do not usually change their shape in a significant way from one section to the following one.

In the candidate correspondence graph there are no edges between contours in the same section. We can find edges only between contours situated on consecutive sections.

3.1.2 Processing – Calculate Minimum Spanning Tree

The minimum spanning tree of the candidate graph can be computed either by using Kruskal's method or by Prim's algorithm. This method can be seen as global, as it examines edges from all over the model, not just from a local situation between a single pair of sections. It constructs the edges that have a greater certainty of inclusion in the component first, before trying to add the less likely ones. This approach does not concentrate on building one component of the object at a time, but processes the edges with the higher weights last, so that the edge that completes the MST is the most likely to be removed by the postprocessing stage.

3.1.3 Postprocessing

The earlier stage has separated those components that are topologically inside other components, or ones that contains other components. For an object with more than one component, the MST calculation always makes erroneous connections between topologically disjoint components. The postprocessing stage presented here isolates each separate component by removing edges from the current graph to give a final correspondence graph.

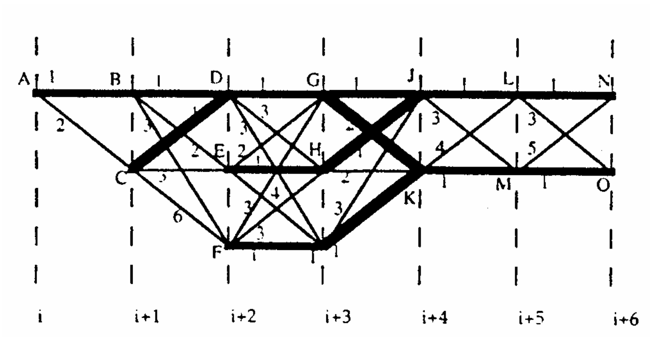

Figure 2
Minimum spanning tree (thicker lines) from a candidate correspondence graph

(a) (b)

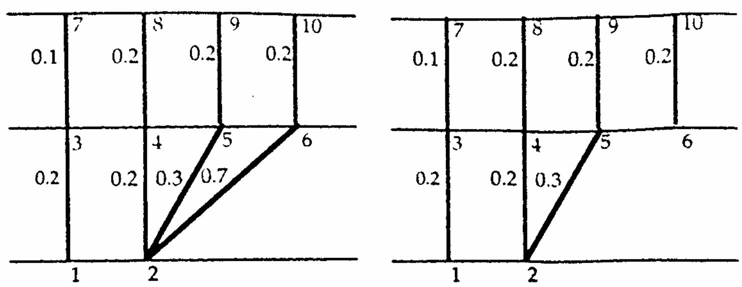

Figure 3
Removing edges during postprocessing (a) – before postprocessing – examine connections between section i and section i+1; (b) – remove edge 2-6, as increase in weight from edge 2-5 is too great.

To eliminate these edges, the postprocessor checks the edges between each pair of sections. If the number of edges is greater than half the number of contours in the two adjacent sections, then an edge included in error may exist. The next step is to examine each of the edges between the two sections. The edges are sorted in order of ascending weight and are examined, starting with the edge with the lowest weight. The next edge in the list is compared to the current edge and if the difference in weight is greater than a threshold value, then all edges with weights greater than the current edge are removed.

3.2 Solutions Based on the Superposition of the Contours

We present two different solutions based on the superposition of the contours. The first one uses the exact value of the area of the intersection. It is more precise, but also more complex. The second one uses an estimation of the superposition degree of the two contours, requiring less computation and which usually is sufficient in most cases.

3.2.1 Computing the Intersection Area between Two Contours

The superposition area of two contours (C_i and C_j) can be defined as follows:

$$A_{i,j} = \text{Area}(\text{Int}(C_i) \cap \text{Int}(C_j)) \quad (3)$$

where Int (Contour) is the interior of the contour and Area(P) is the area of the polygon P. Using the superposition area, the superposition degree $I_{i,j}$ can be defined as follows:

$$I_{i,j} = A_{i,j} / 2A_i + A_{i,j} / 2A_j \quad (4)$$

where A_i is the area of C_i and A_j is the area of C_j. This is the normalization of the superposition area ranging from 0 to 1. If any two contours in the adjacent slices have exactly the same shape, location and orientation, the superposition degree becomes 1. If the two contours are separated, it becomes 0. Only when the superposition degree of a pair of contour is larger than a proper threshold, the contours should be connected.

3.2.2 Estimation of the Superposition Degree of Two Contours

The superposition percentage value of two contours C_i and C_j can be evaluated by using an approximation of this area that requires less computation and it is sufficient in most applications.

We define $xmin_i$, $xmax_i$, $ymin_i$, $ymax_i$, respectively $xmin_j$, $xmax_j$, $ymin_j$, $ymax_j$, the smallest and largest abscissa (respectively, ordinates) of the point belonging to C_i and C_j.

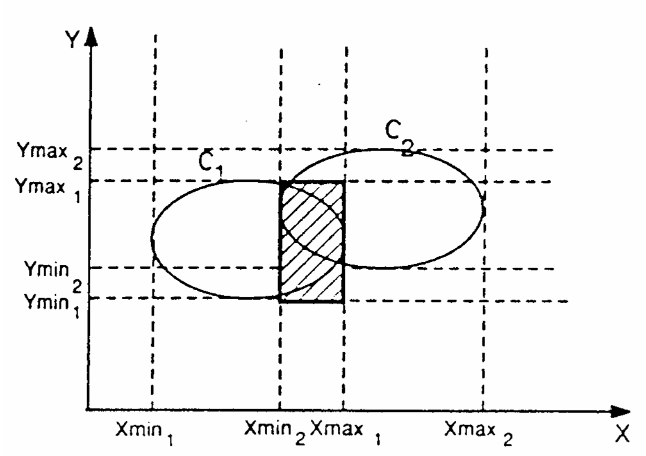

Figure 4
Estimation of superposition degree of two contours

This set of coordinates defines two rectangles encompassing the contours, whose sides are parallel to the x and y-axis. The superposition degree of the two contours will be defined simply from the intersection area:

$A_{i,j} = \mathrm{Inf}([xmax_j - xmin_i], [xmax_i - xmin_j])$

$\mathrm{xInf}([ymax_j - ymin_i],[ymax_i - ymin_j])$ (5)

where $[u] = u$, if $u \geq 0$ and 0, if $u<0$

The superposition degree can be given by the same formula:

$I_{i,j} = A_{i,j} / 2A_i + A_{i,j} / 2A_j$

For both situations, we assume that the two contours are situated on different but adjacent sections.

4 Experimental Results and Performance Comparisons between the Methods of Solving the Correspondence Problem

A set of experiments was made in order to compare the performances of the two algorithms. The tests were made on a series of level curves, increasing the number of curves given as input. The results were put into tables, following these aspects:

- For the graph-based algorithm, tests for determining the optimal threshold used during the post-processing phase;

- For the surfaces algorithm, tests for determining the optimal superposition threshold
- Comparisons between the execution times and necessary corrections for each algorithm.

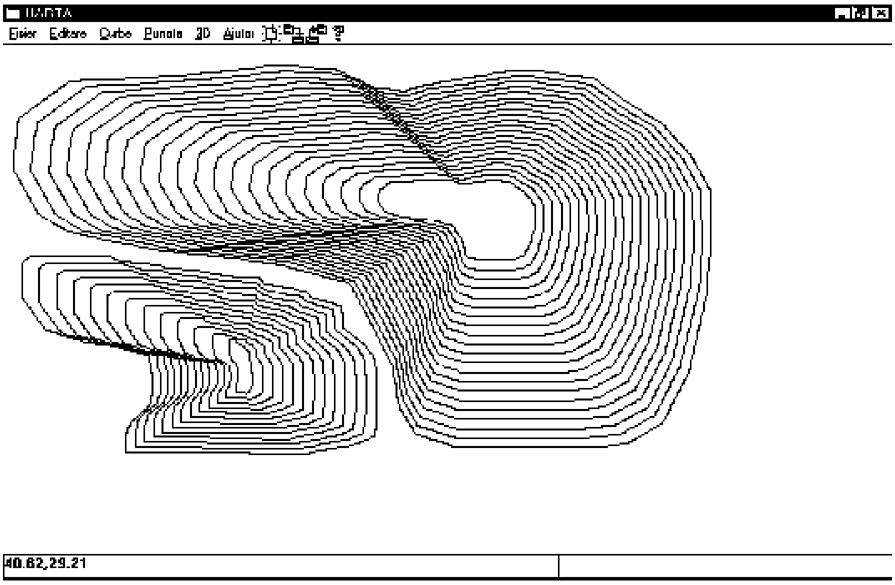

Figure 5

The input of the reconstruction problem

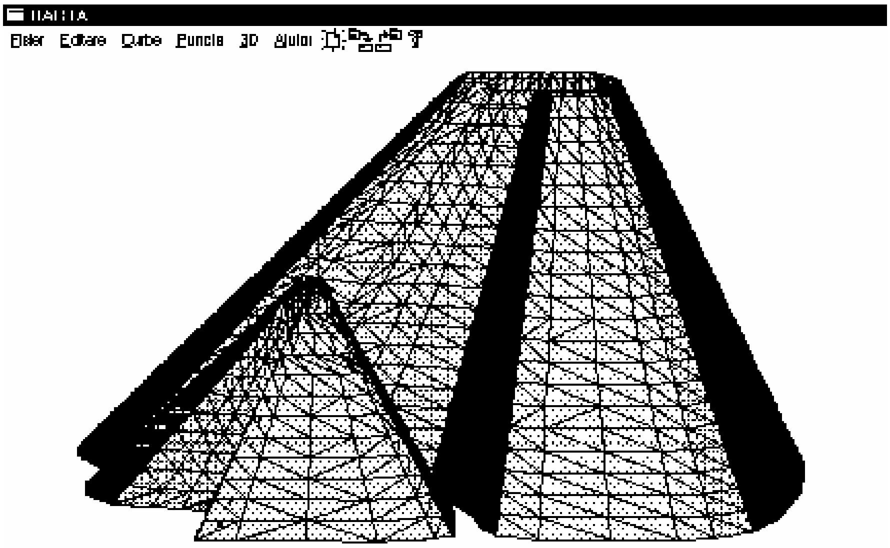

Figure 6
The output of the reconstruction problem

Post-processing threshold	Number of corrections (190 level curves)
0.95	4
0.96	3
0.97	3
0.98	3
0.99	2
1	2
1.01	3
1.02	3
1.04	3
1.05	4

Figure 7
Number of user necessary corrections, function of the post-processing threshold

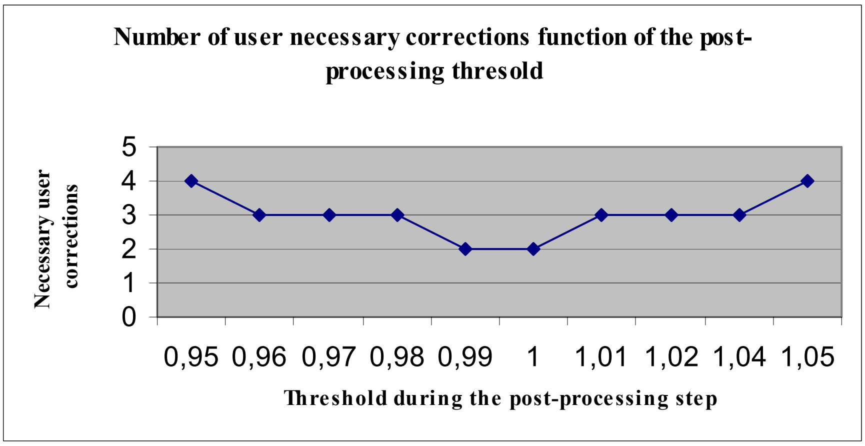

Figure 8
User-corrections graphic, function of the post-processing threshold

Superposition threshold	Number of corrections (190 level curves)
0.85	9
0.86	5
0.87	3
0.88	2
0.89	2
0.90	1
0.91	2
0.92	3
0.93	5
0.94	8
0.95	10

Figure 9
Number of user necessary corrections, function of the superposition threshold

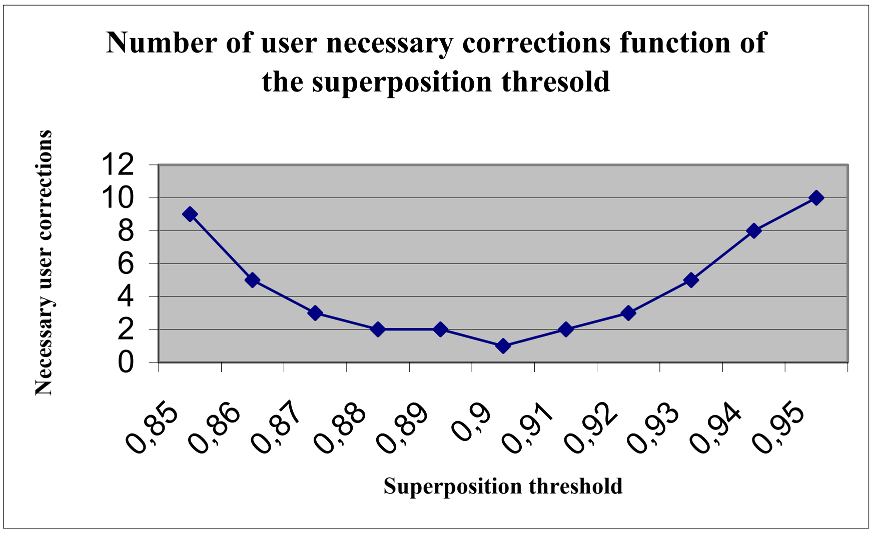

Figure 10
User-corrections graphic, function of the superposition threshold

Nr. of curves	Nr. of resulted 3D entities	Execution time (seconds)		Număr corecții utilizator necesare	
		Graph-based algorithm	Superposition algorithm	Graph-based algorithm	Superposition algorithm
9	1	<0.01	0.01	0	0
20	1	<0.01	0.03	0	0
34	2	0.04	0.13	0	0
54	3	0.06	0.25	0	0
63	4	0.09	0.26	0	0
80	5	0.25	0.48	0	0
116	6	0.83	0.85	0	0
137	7	2.13	2.26	1	0
159	8	3.79	4.27	1	0
190	9	7.49	11.36	2	1

Figure 11
Execution times function of the number of curves

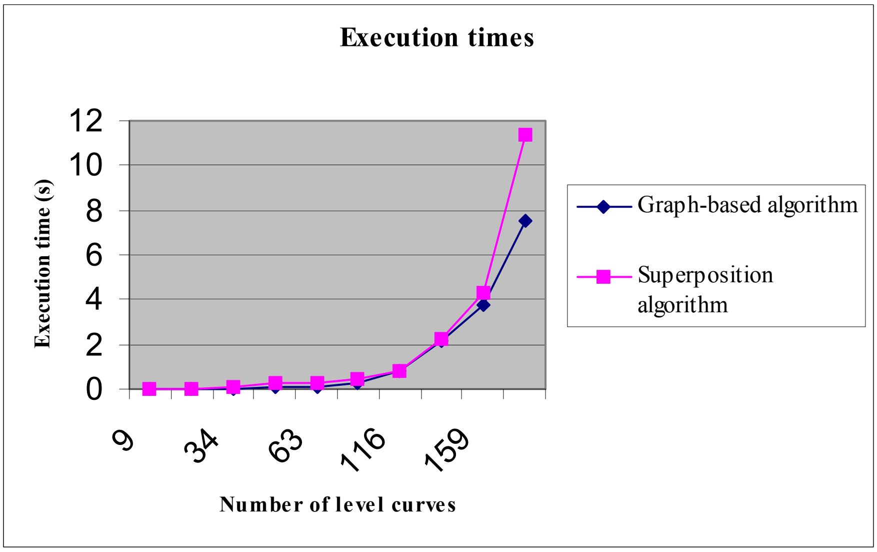

Figure 12

Execution times function of the number of curves

Conclusions

- For the graph-based algorithm, the threshold chosen during the post-processing step must be close to 1 (an edge should not represent more than the double of its predecessor)
- In order to get the best results for the superposition algorithm, the superposition threshold must be close to 0.9
- The execution times are shorter for the graph-based algorithm. The geometrical operations with polygons need more execution time.
- The superposition algorithm needs less user intervention than the graph-based one (meaning connected level curves that the program does not detect).

References

[1] Choi, Y. K., K. H. Park (1994), A heuristic triangulation algorithm for multiple planar contours using an extended double branching procedure, The Visual Computer, No 10, 372 – 387

[2] Ekoule, A. B., F. C. Peyrin, C.L.Odet (1991), A Triangulation Algorithm from Arbitrary Shaped Multiple Planar Contours, ACM Transactions on Graphics, No 10, 182 – 199

[3] Ganapathy, S., T. Dennehy (1982), A new general triangulation method for planar contours, Computer Graphics, No 16, 69-75

[4] Meyers, D., S. Skinner, K. Sloan (1992), Surfaces from contours, ACM Transactions on Graphics, No 11, 228 – 258

[5] Petrescu, Ş., C. Tudose, Z. Racoviță (2000), Reconstruction of three-dimensional objects from parallel planar sections – analysis, solutions and implementation of correspondence problem, University "Politehnica" of Bucharest Scientific Bulletin, No 2, 69-78

[6] Shantz, M. (1981), Surface definition for branching contour defined object, Computer Graphics, 15, 242-270

[7] Tudose, C., Z. Racoviță, Ş. Petrescu (2000), Reconstruction of three-dimensional objects from parallel planar sections – analysis, solutions and implementation of tilling problem, University "Politehnica" of Bucharest Scientific Bulletin, No 1, 63-71