Shape Representation Using Trihedral Mesh Projections

Lluís Ros[1], Kokichi Sugihara[2], and Federico Thomas[1]

[1] Institut de Robòtica i Informàtica Industrial,
CSIC-UPC. Llorens Artigas 4-6, 08028 Barcelona, Spain,
llros@iri.upc.es,fthomas@iri.upc.es,
[2] Dept. of Mathematical Engineering and Information Physics,
University of Tokyo. 7-3-1, Hongo, Bunkyo-ku, Tokyo 113-8656, Japan.
sugihara@simplex.t.u-tokyo.ac.jp.

Abstract. This paper explores the possibility of approximating a surface by a trihedral polygonal mesh plus some triangles at strategic places. The presented approximation has several attractive properties. It turns out that the Z-coordinates of the vertices are completely governed by the Z-coordinates assigned to four selected ones. This allows describing the spatial polygonal mesh with just its 2D projection plus the heights of four vertices. As a consequence, these projections essentially capture the "spatial meaning" of the given surface, in the sense that, whatever spatial interpretations are drawn from them, they all exhibit the same shape, up to some trivial ambiguities.

1 Introduction

A *polygonal mesh* is a piecewise linear 2-manifold made up with planar polygonal patches, glued along the edges, and possibly containing holes. A *polygonization method* is an algorithm able to construct a polygonal mesh approximating a given surface. The literature on polygonization methods, mainly on triangulations, is vast (see [3] for a recent survey on triangulations and algorithms to simplify them). In general, the main goal is to obtain meshes that are close to the surface within a known error, as a way to understand and represent the surface shape [7]. Other goals have been to increase the speed of polygonization and the ability of the polygonizer to satisfy some constraints in the solution (e.g., one might request the most accurate approximation using a given number of line segments or triangles).

In general, a polygonal mesh cannot be reconstructed from its projection onto a plane because infinitely many meshes generate exactly the same projection. For example, for the triangular mesh projection in figure 1, there are many different reconstructions, as illustrated. The first two seem to have no meaning; but, actually, there is a rather "hidden" meaningfull reconstruction: Nefertiti's face! Can we obtain a spatial mesh approximating Nefertiti's face in such a way that its projection still keeps its spatial meaning?

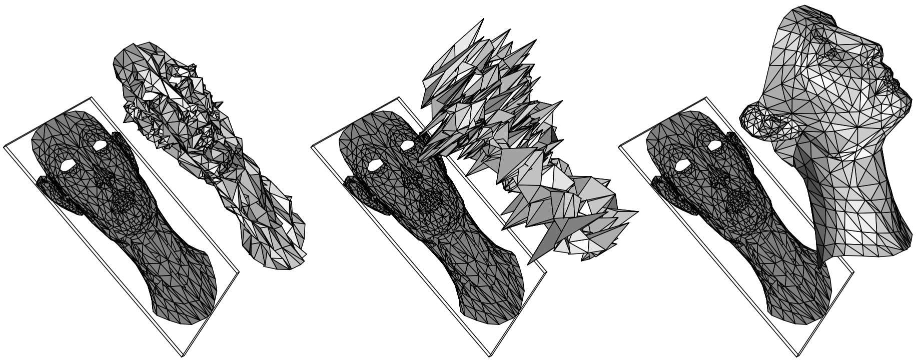

Fig. 1. Arbitrary reconstructions of this triangulated projection have no spatial meaning. But actually, a very specific one of them really does: it shows Nefertiti's face.

There is a class of meshes whose projections fully determine the spatial shape once the heights of four vertices are given. We call these projections *unequivocal* because their reconstructions represent essentially the same object, up to some trivial ambiguities. For example, the projection in figure 2a unequivocally represents a truncated tetrahedron, as seen in figures 2d, e, and f. Observe that it suffices to set the heights of P, Q, T and R to determine those of S and U, using the fact that all cofacial vertices must be coplanar and, hence, S must lie on the face-plane $RPQS$, and U on $SQTU$.

One of our goals is then to approximate any given surface with a polygonal mesh yielding unequivocal projections that uniquely identify the spatial shape up to the trivial ambiguities produced by changing the heights of only four vertices. Section 2 presents the trihedral polygonal mesh, the model we use to this end, and shows how its projections are unequivocal in the sense given above.

Nevertheless, we need to go beyond this goal if this representation is to be useful. Consider what happens if the (x, y) vertex positions in figure 2a are slightly altered (figure 2b). The new projection no longer represents a correct truncated tetrahedron for, to be so, the edges joining the two triangular faces, when extended, should be concurrent at the apex of the (imaginary) original tetrahedron. Equivalently, note that once P, Q, R and T are given, the height of U is overconstrained, for it can be calculated from *both* the coplanarity of $SQTU$ or that of $RPTU$. For generic vertex positions, the two values of this height do not necessarily coincide, and the only spatial reconstruction that keeps cofacial vertices coplanar is a *trivial* one, with *all* vertices lying on a single plane [5, 6]. This makes the four provided heights inconsistent between each other. In sum, the consistency of the four heights only holds at very specific positions of the vertices and inevitable discretization errors will make this representation useless. This problem is common in Computer Vision [8] and Computer Graphics [12, 10], and mathematical characterizations of generically consistent projections are given in [11, 9]. The way we use to make this representation robust against these

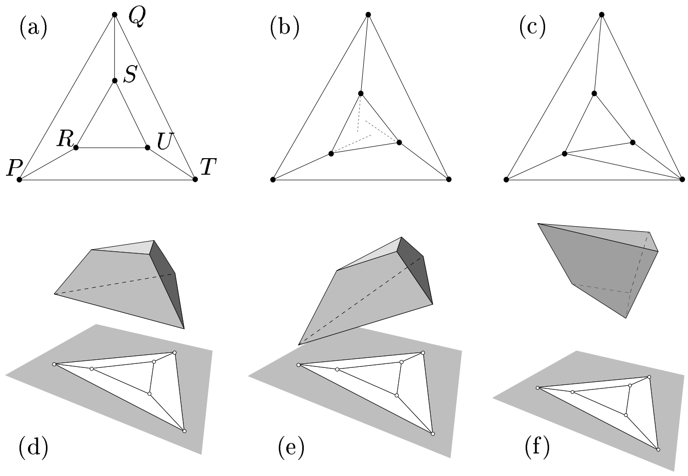

Fig. 2. A truncated tetrahedron (a) and three possible reconstructions (d, e, f). The slightest perturbation destroys the correctness of the projection (b), but this can be avoided adding new triangular faces (c).

errors follows from this observation: if the height of a vertex in a projection is overconstrained because the vertex lies on several planes that fix it, we just introduce new triangular faces around it for preventing this to occur (figure 2c). Section 3 gives a fast algorithm to this end, derived from this observation, using the so-called T/TT-transformations. Section 4 describes a complementary optimization step that properly places these transformations to minimize the reconstruction errors by reducing the problem to a cyclic AND/OR graph search. We finally conclude in section 5.

2 Trihedral Polygonal Meshes

Trihedral meshes, i. e., those where all vertices have exactly three incident faces, produce unequivocal projections. Indeed, figure 3 shows that in them, after fixing the planes of two adjacent faces, we have enough data to derive the heights of the remaining vertices. Clearly, the heights of the bold vertices fix the shadowed face-planes and the heights of other vertices on them. At this point, any other surrounding face has three vertices whose height is known and, so, its plane can be fixed too. The same argument can be iteratively applied and the result is a *height propagation* reaching all vertices in the projection.

In the schematic representation of this height propagation (figure 3) every face f receives three incoming arrows from the three vertices that fix it. The derivation of heights for the rest of vertices on f is indicated with outgoing arrows from f. The result is a tree-shaped structure spanning all vertices and faces. In this tree, a path from any of the initial four vertices to any other vertex will be hereafter referred to as a *propagation wave*. Note that, height propagations where a face is fixed from three (almost) collinear vertices must be avoided. Section 4 gives a way to compute propagations eluding these collinearities.

A trihedral mesh approximating a convex or concave surface can be readily obtained by distributing a set of random points all over the surface and computing its tangent planes at these points. This leads to a plane arrangement whose upper envelope –if the surface is convex– or lower envelope –if it is concave– provides a good mesh approximation of the surface. Since the tangent plane orientations are random, any three of such planes meet in a single point, and hence the mesh is trihedral.

Alternatively, a trihedral mesh approximation of a piece of concave or convex surface can be obtained by starting with a rough mesh approximation and iteratively applying a bevel-cutting [2] and/or a corner-cutting [1] operation to attain the desired approximation.

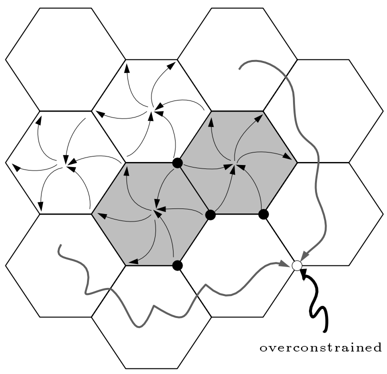

Fig. 3. A height propagation starting at four pre-specified (bold) vertices. Several vertices can have an overconstrained height.

Obviously, the situation becomes much more complex when concavities and convexities are simultaneously present. The first step in these cases would be to decompose the surface into patches having congruent signs for the maximum and minimum curvatures at all their points. If this is done for a general C^∞ surface, we would get patches labeled $(+,+)$, $(+,-)$, $(-,+)$ or $(-,-)$ separated by curves which could be labeled with $(-,0)$, $(+,0)$, $(0,-)$, or $(0,+)$, and isolated points (actually, maxima or mimima) which would be labeled with $(0,0)$. Saddle points would be also labeled with $(0,0)$ but they would appear as intersections of separating curves. If we extend this treatment to C^2 surfaces, we could get entire patches with one of the above nine possible labels. For example, all plane patches would have the label $(0,0)$.

Patches labeled with $(+,+)$ or $(-,-)$ represent fully convex or concave patches and thus they can be polygonized as described above. Patches labeled as $(*,0)$ or $(0,*)$ can be polygonized by locating random points along the direction of maximum curvature. Patches $(0,0)$ would only require a single point on them. Unfortunately, the treatment of $(+,-)$ or $(-,+)$ patches remains as an open problem for us.

The connection between polygonized patches can be obtained by computing tangent planes on points along their common boundaries. In sum, the polygonization we propose can be done, first for each patch by generating the tangent planes in a sufficiently high density, and next by connecting them using the tangent planes generated along their common boundaries.

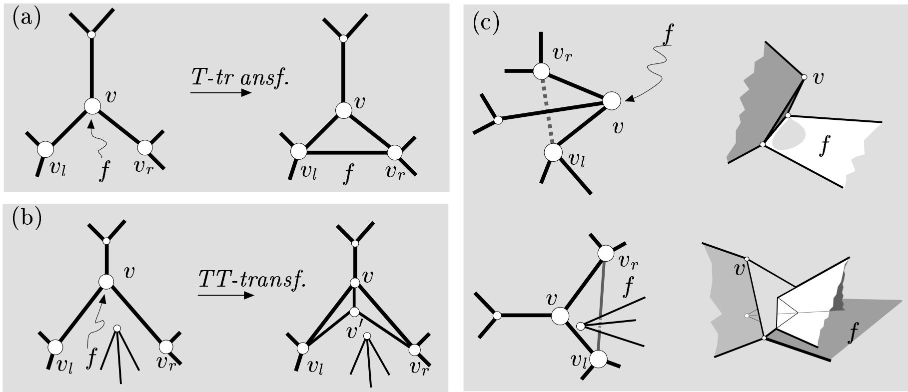

Fig. 4. (a and b) T and TT-transformations. (c) Overhanged and self-intersecting reconstructions induced by T-transformations at locally non-convex faces.

3 T and TT-Transformations

In a trihedral mesh a projection is overconstrained because any of its vertices lies on three faces and, potentially, up to three propagation waves can determine a height at the same time. However, as done in figure 2c, this can be avoided by adding triangular faces. To this end, we first compute an arbitrary height propagation spanning all vertices, and check which of them receives more than one wave. We then take one overconstrained vertex v at a time and prevent all *but one* waves from reaching v as follows. To stop the wave getting v from face f, we apply either of these two transformations (figure 4a and b):

- A *T-transformation*, which places a new edge joining the two neighboring vertices of v in f, say v_l and v_r.
- A *TT-transformation*, which places a new vertex v' on f near v and the three new edges (v', v), (v', v_l) and (v', v_r).

After either transformation, f cannot constrain the height of v anymore. Also, the added triangles are innocuous because all heights can still be determined from the four initial ones.

Which transformation is preferred depends on the geometry of face f around vertex v. If all points inside the triangle $v_l v_r v$ belong to f, we say that f is *locally convex* at v. So, for situations where f is locally convex at v, simplicity prevails and T-transformations are enough (figure 4a). When local non-convexities are present (figure 4b), T-transformations yield occluded or partially occluded crossing edges whose spatial reconstructions have overhanged parts, or self-intersecting faces (figure 4c). Here, TT-transformations are preferred for they can avoid this.

An observation complements the strategy. In an overconstrained vertex v, either two or three incoming propagation waves arrive. If *no more* than one of

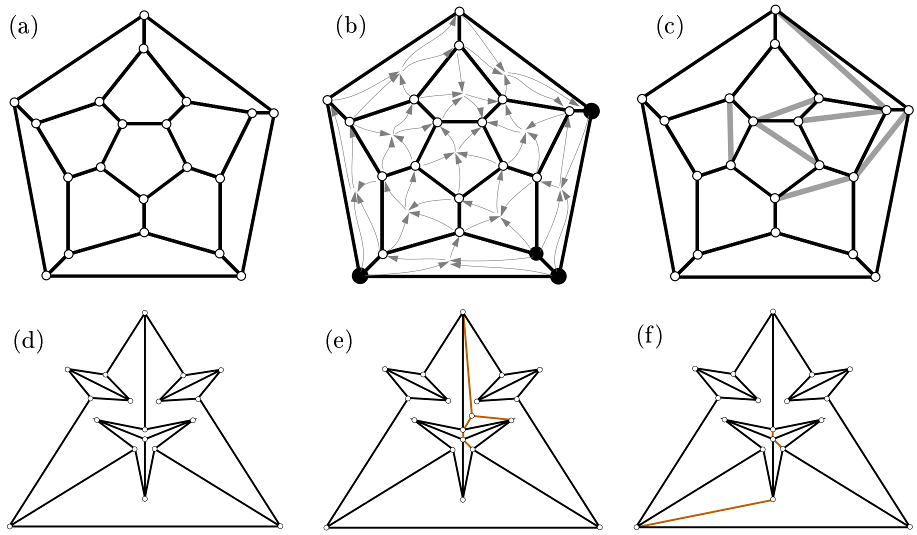

Fig. 5. A projected dodecahedron (a) together with a height propagation (b) and the T-transformations it yields (c). A protruded tetrahedron (d) and two possible corrections: (e), involving TT-transformations, and (f), involving only T-transformations.

them comes through a locally non-convex face, then we can always drop the incidence constraint in this vertex just with T-transformations: we just leave the eventual "bad" wave to determine the height of v and stop the others with T-transformations. This completes the description of a one-sweep algorithm removing overdetermination. As an example, figures 5a-c show a projected dodecahedron before and after applying T-transformations.

In general, when the approximated surface is uniformly convex, or uniformly concave, all faces of the resulting trihedral polygonal mesh will be locally convex, and hence T-transformations will suffice. However, even when local non-convexities exist at the faces, there still might be some height propagations where only T-transformations suffice. In figure 5e, for example, an algorithm computing an arbitrary propagation can be forced to use TT-transformations, whereas with a proper search, a robust projection is obtained only with T-transformations (figure 5f). But one certainly finds correct projections where no propagation strictly using *T-transformations* can be found [5, Section 8.4].

4 Optimal Propagations and Cyclic AND/OR Graphs

The algorithm in the preceeding section corrects the incidence structure by finding an arbitrary height propagation and inserting a T or a TT-transformation whenever a vertex height is determined by two or more faces. However, arbitrary propagations might travel along "degenerate paths" where the planes for some of the faces are determined by three aligned (or almost aligned) vertices. Clearly, these *degenerate propagations* must be avoided if we want to minimize the errors during the reconstruction of the spatial shape from the initial set of four heights. This section provides an algorithm to find height propagations that avoid these degeneracies by formulating the problem as that of finding the least cost solution of a cyclic AND/OR graph [4]. We now recall some preliminary concepts about this kind of graphs.

An AND/OR *directed* graph G, can be regarded as a hierarchic representation of possible solution strategies for a major problem, represented as a *root node*, r, in G. Any other node v represents a subproblem of lower complexity whose solution contributes to solve the problem at hand.

There are three types of nodes: AND nodes, OR nodes and TERMINAL nodes. Every node v has a set $S(v)$ of *successor nodes*, possibly empty, to which it is connected in either of two ways:

- An AND node v is linked to all nodes $s_i \in S(v)$ through directed AND arcs (v, s_i), meaning that the subproblem for v can be trivially solved once *all* subproblems for the nodes in $S(v)$ have been solved.
- An OR node v is linked to all nodes $s_i \in S(v)$ through directed OR arcs (v, s_i), meaning that the subproblem for v can be trivially solved once *any one* of the subproblems for the nodes in $S(v)$ has been solved".
- A TERMINAL node represents a yet-solved or trivial subproblem and has no successors.

With this setting, a *feasible solution* to the problem becomes represented as a directed subgraph T of G verifying:

- r belongs to T.
- If v is an OR node and belongs to T, then exactly one of its successors in $S(v)$ belongs to T.
- If v is an AND node and belongs to T, then every successor in $S(v)$ belongs to T.
- Every leaf node in T is a TERMINAL node.
- T contains no cycle, it is a tree.

One can also assign a cost $c(u, v) > 0$ to every arc (u, v) in G and ask for the solution T with minimum overall cost $C(T) = \sum_{(u,v)\in E(T)} c(u, v)$, where $E(T)$ is the set of arcs of T. Note that, as defined, G can contain cycles. This turns out to be the main difficulty for this optimization problem, which, in the past, was usually tackled by a rather inefficient trick: "unfolding" the cycles and applying standard AND/OR search methods for acyclic graphs. However, explicit treatment of cycles has recently been considered, and an efficient algorithm is achieved in [4].

The search for an optimal height propagation is next reduced to this model. This amounts to (1) constructing an AND/OR graph G_{hp} whose feasible solutions define a height propagation, and (2) define a cost function that promotes non-degenerate propagations over degenerate ones.

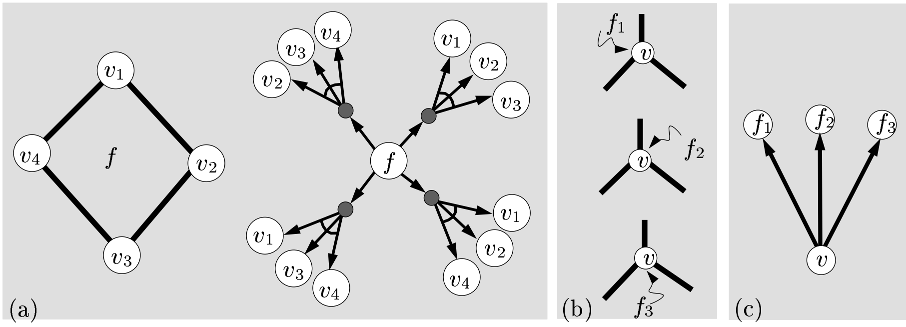

Fig. 6. AND/OR subgraphs for the propagation rules. AND nodes are indicated by joining all their emanating arcs. (a) Constructed subgraph translating rule **R2** for a quadrilateral face. Dummy-face nodes are shadowed in grey. Note that, actually, there is only one vertex node for each vertex in the trihedral mesh, but for clarity they are here duplicated. (b) Propagation waves reaching a vertex. (c) Subgraph for rule **R3**, with an arc for each of the possibilities in (b).

4.1 Feasible Height Propagations

A height propagation can be defined by the following rules, with the given straightforward translation into AND/OR subgraphs.

R1: *Four selected vertices of the projection trigger the propagation.* For this, we put a TERMINAL node for each of the triggering vertices.

R2: *Every face in the polygonization can be determined once the heights of any three of its vertices are determined.* If $deg(f)$ denotes the number of vertices of face f, then there are $c_f = \binom{deg(f)}{3}$ possible combinations of three vertices determining f. If we put a node in G_{hp} for every vertex, except for the four triggering ones, then this rule is translated by adding an OR node for every face, linked to c_f new "dummy-face" AND nodes, each representing one of the above combinations. Each dummy-face node is in turn linked with arcs to the three involved vertices in the combination. Figure 6 gives a schematic representation. The newly introduced vertex nodes have not been assigned a type yet. This type is induced by the following rule.

R3: *Except for the initial four verties, the height of every other vertex is determined once one of its incident faces has a determined plane.* This implements the fact that the propagation wave fixing the height of a vertex can come from any of its three incident faces (figure 6b). This rule can be represented by setting each vertex node as OR type, and linking it to the face nodes of its incident faces figure 6c.

R4: *The height propagation must reach all vertices.* For this, we add a root AND node r to G_{hp} and link it to all vertex nodes.

Note that a feasible solution tree of G_{hp} provides instructions to derive a height propagation that reaches all vertices, starting at the four pre-specified heights.

4.2 Cost Function

In order to penalize propagations using sets of almost-aligned vertices, we proceed as follows. Consider a height propagation that fixes a face-plane f from the point coordinates of three previously fixed vertices v_i, v_j and v_k. We can simply penalize the corresponding arcs in G_{hp} emanating from f by giving them a cost that is inversely proportional to the area of the triangle defined by v_i, v_j and v_k in the projection. The rest of arc costs are actually irrelevant, but need to be positively defined [4]. In sum, for every directed arc (u, v) we define its cost as follows:

1. $c(u, v) = 1/det(v_1, v_2, v_3)$, if u is a dummy-face AND node and v is any one of its descendants. Here, v_i, v_j and v_k are the homogeneous coordinates of the vertices associated with the three descendants of u.
2. $c(u, v) = 1$, if u is an OR node.
3. $c(u, v) = 1$, if u is the root AND node.

Once the least cost solution T is found, the projection can be made robust to slight vertex perturbations as follows. At a vertex v receiving more than one propagation wave, we put a T/TT-transformation on all faces fixing v, except on the one in the propagation wave represented in T.

4.3 Complexity Analysis

The worst-case complexity of computing the optimal solution of a cyclic AND/OR graph with n nodes is $O(n^3)$ [4]. We now prove that the number of nodes in G_{hp} grows linearly with the number of vertices of the trihedral polygonal mesh.

Let e, v and f be the number of edges, vertices and faces of the given mesh. Then, $2e = 3v$ because the mesh is trihedral. Moreover, if the mesh has h holes, with “the outside” of the mesh counting as a hole too, then Euler's relation says that $v - e + f = 2 - h$. From these two equalities the number of faces of the mesh can be written in terms of the number of vertices and holes, $f = \frac{v+4}{2} - h$. Let us now count the number of nodes added by each of the rules **R1**,...,**R4**:

- Rule **R1** adds four vertex nodes.
- Rule **R2** adds one OR node for each face, amounting to $f = \frac{v+4}{2} - h = O(v)$ total nodes, assuming a constant number of holes. Also, for every face f this rule adds $c_f = \binom{deg(f)}{3}$ dummy-face AND nodes. Although this number is clearly in the worst case $O(deg(f)^3)$, if we divide the sum of face degrees by the number of faces, the average face degree is six, at an increasing number of randomly placed vertices in the mesh:

$$\frac{\sum_{allfaces} deg(f_i)}{f} = \frac{3v}{\frac{v+4}{2} - h} = \frac{6v}{v + 4 - 2h} \;,$$

  which will keep the number of dummy-face AND nodes linearly growing:

$$\binom{6}{3} f = 20 \left(\frac{v+4}{2} - h \right) = O(v).$$

- Rule **R3** adds a linear number of OR vertex nodes.
- Rule **R4** only adds one AND node, the root.

Up to now we have assumed that the four vertices triggering the propagation are a priori selected. But other height propagations starting at other four vertices could yield better height propagations. To test all possibilities, we do not need to repeat the AND/OR search for every different combination of four vertices. Indeed, note that these vertices just fix the planes of the faces they belong to. So, any other set of four vertices on these faces will yield the same optimal propagations, provided that two of them lie on the common edge. We can equivalently think of pairs of faces triggering the propagation and use their face nodes as TERMINAL in G_{hp}. The choice of TERMINAL vertices (instead of TERMINAL faces) was done to be coherent with previous explanations. In sum, if one wants to search over all possible starting places of propagation, then for each pair of adjacent faces the AND/OR search needs to be repeated. This amounts to solve $e = \frac{3}{2}v$ optimization problems in the worst case, meaning that the overall complexity will be $O(v^4)$, under the assumption that the face degree is six.

5 Conclusion

We have shown how trihedral mesh projections can capture the spatial shape of a given object's surface, up to some trivial ambiguities. We have also presented a local strategy that takes a trihedral projection as input and places some triangular faces at strategic places until it is made robust to perturbations in its vertex coordinates. Finally, we have found how to put these triangles so that the spatial reconstruction is performed in the most accurate way possible, avoiding height propagations along degenerate paths.

Although we can deal with an important range of surfaces, no algorithm has been devised yet to obtain trihedral meshes approximating surfaces with saddle-crests or saddle-valleys. This constitutes a main issue for further research.

Acknowledgements

The authors wish to thank Carme Torras and Pablo Jimenez for fruitful discussions around the AND/OR search strategy.

This work has been partially supported by the Spanish Ministry of Education under grant FPI94-46634232, by the Spanish CICYT under contracts TIC96-0721-C02-01 and TIC2000-0696, and by the Grant-in-Aid for Scientific Research of the Japanese Ministry of Education, Science, Sports and Culture.

References

1. C. Boor. Cutting corners always works. *Computer Aided Geometric Design*, 4:125–131, 1987.

2. D. Fox and K. I. Joy. On polyhedral approximations to a sphere. In *IEEE Int. Conf. on Visualization*, pages 426–432, 1998.
3. P. Heckbert and M. Garland. Survey of polygonal surface simplification algorithms. In *Multiresolution Surface Modeling Course SIGGRAPH'97*, May 1997. Available at `http://www.cs.cmu.edu/~ph`.
4. P. Jiménez and C. Torras. An efficient algorithm for searching implicit AND/OR graphs with cycles. *Artificial Intelligence*, 124:1–30, 2000.
5. L. Ros. *A Kinematic-Geometric Approach to Spatial Interpretation of Line Drawings*. PhD thesis, Polytechnic University of Catalonia, May 2000. Available at `http://www-iri.upc.es/people/ros`.
6. L. Ros and F. Thomas. Overcoming superstrictness in line drawing interpretation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*. Accepted for publication. Preliminary scheduled for Vol 24, No. 4, April 2002.
7. W. Seibold and G. Wyvill. Towards an understanding of surfaces through polygonization. In *IEEE Int. Conf. on Visualization*, pages 416–425, 1998.
8. K. Sugihara. An algebraic approach to shape-from-image problems. *Artificial Intelligence*, 23:59–95, 1984.
9. K. Sugihara. *Machine Interpretation of Line Drawings*. The MIT Press, 1986.
10. K. Sugihara. Resolvable representation of polyhedra. *Discrete and Computational Geometry*, 21(2):243–255, 1999.
11. W. Whiteley. Some matroids on hypergraphs with applications to scene analysis and geometry. *Discrete and Computational Geometry*, 4:75–95, 1988.
12. W. Whiteley. How to design or describe a polyhedron. *Journal of Intelligent and Robotic Systems*, 11:135–160, 1994.