Exploiting 2D Topology in Labeling Polyhedral Images

Van-Due Nguyen

General Electric Corporate Research and Development Center
Schenectady, NY 12301

Abftract: A polyhedral image is a segmentation of the image plane into connected regions, called faces, joined by vertices and edges. The segmentation is represented by a planar network of nodes (vertices, edges, faces) linked by adjacency links. The labeling constraints at a node are all local labelings of the node consistent with itself and all its adjacent neighbors. The local labelings are represented by junctions, junction-pairs, and junction-loops respectively for the vertices, edges, and face boundaries of the image.

Constraint satisfaction and propagation is done uniformly over all nodes in the image, from each node to its adjacent neighbors. The result is local consistency or inconsistency at all the nodes in the planar network. We show that globally consistent labelings of the image exist, if and only if all the nodes in the network have locally consistent labelings. The planar network of nodes, with labels and local labelings attached to each node, represents all locally/globally consistent labelings of the polyhedral image.

1. A Labeling Example

Figure 1 traces the parallel labeling of two blocks, one on top of the other. The parallel labeling starts with the input image and default labels (?) for all the nodes (i.e., vertices, edges, and faces) in the image. Then, it finds all local labeling constraints at all vertices, edges, and face boundaries of the image, described respectively by junctions, junction-pairs, and junction-loops, frames 1 to 3. Attached to each node is the number of local labelings. Constraint satisfaction and propagation (CSP) is done uniformly at all nodes in the image, from every node to its neighbors, through vertex-edge, edge-face, and face-vertex links. Note the decrease in the number of local labelings at each node, frames 4 to 6.

The face surrounding the two blocks has 16 local labelings, corresponding to the blocks floating in air, or resting against some imaginary surface at some of its bounding edges. The two interpretations correspond to the face labeled as image background (B) or as polyhedral face (F). Note that the blocks sitting on top of a horizontal surface can be thought as the blocks floating in air, and infinitesimally touching the surface. The surrounding face is a touchable background, and so is labeled by 8, and has only 1 junction-loop. The detection of the background face surrounding the blocks further restricts the labeling of the blocks, Frames 7 and 8 show the final labels at the edges and faces.

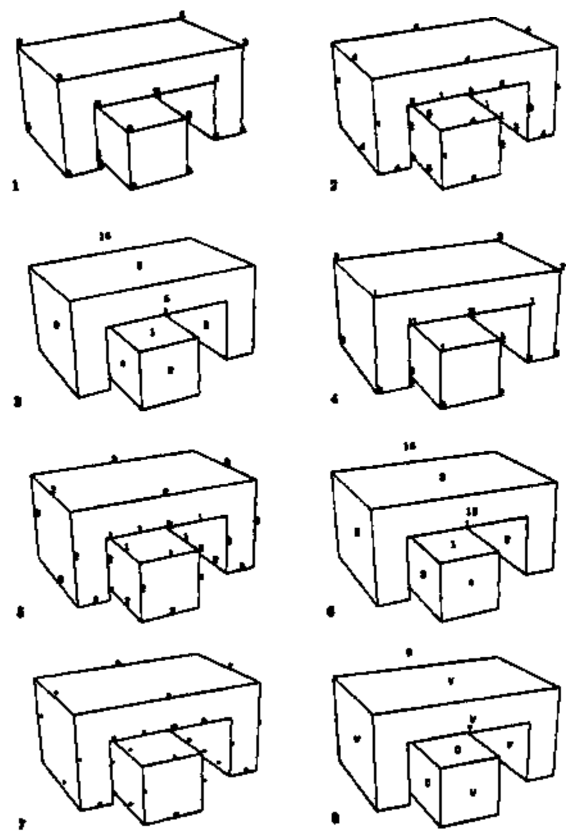

Figure 1: Parallel labeling of the blocks.

2. Image Topology

Just as the vertices, edges, and faces of a 3D polyhedron join 3D blocks (Markowsky and Wesley 1980), the vertices and edges of an image join connected regions of the image, called faces.

Definitions 1 *(Topology in a polyhedral image)*

- *A* vertex *is not only an intersection of distinct edges, but also a junction of distinct faces of the image.*
- *An* edge *is not only an intersection of distinct faces, but also a segment connecting distinct vertices.*
- *A* face *is a connected region of the image plane locally bounded either by chains/cycles of distinct edges, or by open/closed paths going through distinct vertices.*

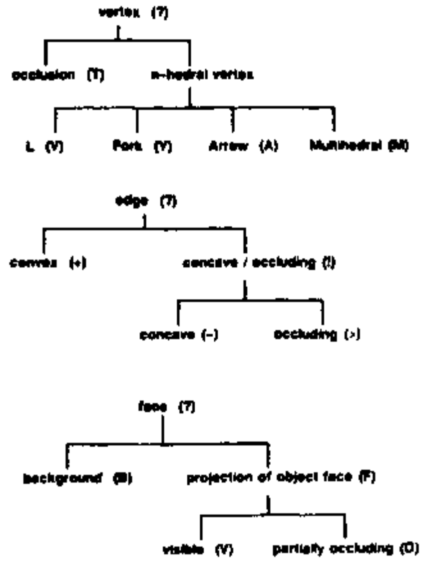

Figure 2: Label hierarchies for vertices, edges, and faces.

The topology of the image as junction/connection/boundary of subsets does not hold only at the image boundary 67, and so ∂I must be excluded from labeling. Nguyen (1987) proved that:

Theorem 1 *The image of a 3D polyhedral scene is always a segmentation of the open subset inside the image boundary ∂I into faces, joined by edges and vertices.*

3. Enumeration of All Local Labelings

Figures 2 show the label hierarchies for the vertices, edges, and faces. Just as object edges and faces can be represented in terms of object vertices, local labelings around image edges and faces are naturally represented as n-tuples of labelings at image vertices. Figures 3 describe the local labelings around a vertex, an edge, and a face. The junctions for occlusion and tri-hedral vertices (Huffman 1971, Clowes 1971) are precomputed and stored in a dictionary. Junctions and vertex-labels are related by a C relation, which is implicit in the dictionary representation. The junctions for multihedral vertices with zero or one hidden face can be enumerated depth-first from the Fundamental Junction Equation relating the orientation of the edges, the ordering of the faces, and their respective gradient points around a vertex (Malik 1985).

The junction-loops of a face *F* are found by a depth-first search for all consistent labelings around the face boundary δF. Consistency means that the junctions of any two consecutive vertices must have the same edge-label for the edge connecting these two vertices. Each junction-loop found from depth-first search must be checked to see if the edge-labels of the boundary edges are all consistent with some face-label of *F*. This later is node-consistency enforced by the $\subseteq$ relation, relating junction-loops and face-labels. Similarly, junction-pairs describe local labelings for edges, and are found by an exhaustive but local search.

The maximum number of edge-labels in the edge hierarchy, mE, is constant and equal to five for polyhedral images. The maximum number of junctions (resp. junction-pairs) at any vertex (resp. edge) of the image, *mj* (resp. mp), is exponential in the complexity of the image, described by the three parameters ($\max|V|$, $\max|E|$, $\max|F|$):

$$m_J = m_E^{\max|V|}$$
$$m_P = m_J^{\max|E|} = m_E^{2\max|V|}$$

Worse than m_J and m_P, the maximum number of junction-loops at any face of the image, m_L, is doubly exponential in the complexity of the image:

$$m_L = m_J^{\max|F|} = \left(m_E^{\max|V|}\right)^{\max|F|}$$

4. Constraint Satisfaction and Propagation

Constraints at all nodes in the planar network should be propagated uniformly over all vertex-edge, edge-face, and face-vertex links. The local labeling constraints are the junctions, junction pairs, and junction-loops. Let's describe the complete *CSV* over vertex-edge-face links, after all the local constraints are found and enumerated.

Algorithm 1 (Parallel CSP over vertex-edge-face links)
Initialise Vertex-Fifo, Edge-Fifo, and Face-Fifo respectively with all vertices, edges, and faces in the image
Loop until (and (emptyp Vertex-Fifo) (ernptyp Edge-Fifo) (emptyp Face-Fifo)) do

1. Loop until (emptyp Vertex-Fifo) do
   Let *V* := (pop Vertex-Fifo).
   a. Loop for all edge E_i in Edges(V) do
      Restrict junction-pairs of E_i with junctions of *V*
      If the set of junction-pairs of edge E_i has been restricted then (insert E_i Edge-Fifo).
   b. Loop for all face F_i in Faces(V) do
      Restrict junction-loops of F_i with junctions of *V*
      If the set of junction-loops of face F_i has been restricted then (insert F_i Face-Fifo).
2. Loop until (emptyp Edge-Fifo) do
   Let *E* := (pop Edge-Fifo).
   a. Loop for all face F_i in Faces(E) do
      Restrict junction-loops of F_i with junction-pairs of *E*.
      If the set of junction-loops of face F_i has been restricted then (insert F_i Face-Fifo).
   b. Loop for all vertex V_i in Vertices(E) do
      Restrict junctions of V_i with junction-pairs of *E*.
      If the set of junctions of vertex V_i has been restricted then (insert V_i **Vertex-Fifo).**
3. Loop until (emptyp Face-Fifo) do
   Let *F* := (pop Face-Fifo).
   a. Loop for all vertex V_i in Vertices(F) do
      Restrict junctions of V_i with junction-loops of *F*.
      If the set of junctions of vertex *Vi* has been restricted then (insert V_i Vertex-Fifo).
   b. Loop for all edge *Et* in Edges(F) do
      Restrict junction-pairs of E_i with junction-loops of *F*.
      If the set of junction-pairs of edge E_i has been restricted then (insert E_i Edge-Fifo).

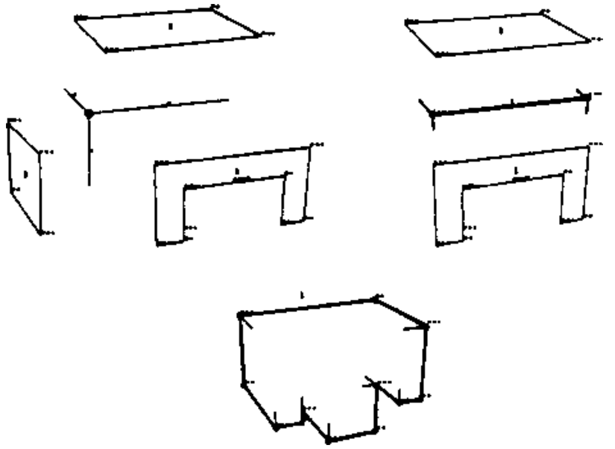

Figure 3: Junctions, junction-pairs, and junction-loops are respectively local labelings around a vertex, an edge, and a face.

Algorithm 2 **(Restrict junctions of V with junction-pairs of E)**
Loop for all J in Junctions(V) do
if Loop for all P in Junction-Pairs(E)
never $J \subseteq aref(P,V)$
then if Loop for all P in Junction-Pairs(E)
thereis $aref(P,V) \subseteq J$
then replace J by its sub-junctions
else delete J from Junctions(V).

Complexity 1 ***The time complexity of constraint satisfaction and propagation over vertex-edge-face links of a polyhedral image I is:***

$$O\left(m_J\, m_P\, m_L \left(\frac{m_P + m_L}{m_J} + \frac{m_L + m_J}{m_P} + \frac{m_J + m_P}{m_L}\right) |Edges(I)|\right)$$

5. Global Consistency

Local consistency are nothing but 1) node consistency between local labelings and labels for all nodes, and 2) arc consistency between two adjacent nodes for all arcs. Node consistency is enforced by $\subsetneq$ relation when we enumerate the local labelings at a node. Arc consistency is enforced both ways by the loops in Algorithm 1. Global consistency from local node and arc consistencies at all vertices, edges, and faces of the image can be proved by labeling inductively from one face to its neighboring faces, covering the whole image. For proof, the reader is referred to (Nguyen 1987).

Let's compare the parallel labeling algorithm with more general results from topology (Hocking and Young 1961, Henle 1979) and from consistency of networks of constraints (Waltz 1975, Montanari 1976, Mackworth 1977, Freuder 1978). Recall that image labeling used to be called line labeling, because the core problem is to label all the lines or edges in the image. The network of edges with links between intersecting edges is not planar. By introducing vertices as new nodes and links between adjacent vertices and edges, the resulting network of edges and vertices is planar. Similarly, we add faces as nodes, and links between faces and their adjacent vertices and edges. The adjacency links describe the boundary relationship *d* between topological nodes. The planar topology of the image guarantees that the network of vertices, edges, and faces is planar.

Junctions are used to capture the n-ary constraints among edges intersecting at a common vertex. They are exhaustively enumerated by looking at a 3D corner from different viewpoints. A vertex-node is needed to store these junctions. The n-ary constraints among intersecting edges become binary constraints between these edges and the common vertex. Similarly, by creating higher level k-nodes and enumerating all the locally consistent labelings for these *k*-nodes, one can synthesize more extended -ary constraints, covering finally all the edges in thek image (freuder 1978).

Labeling an image can be seen as putting together a puzzle where the pieces are the vertices, edges, and faces of the image. The pieces are required to have matched labelings at their interfaces. A vertex has junctions as labelings. An edge must have junction-pairs as labelings because it interfaces with its two ending vertices and its two adjacent faces through vertices. Similarly, a face must have junction-loops as local labelings. The segmentation of the image into vertices, edges, and faces guarrantees that labeling over the whole image is equivalent to enumerating the local labelings at all the pieces, such that they satisfy their respective unary constraints, and doing arc consistency between adjacent pieces, such that they have matched labelings at their interfaces. From another perspective, the 2D topology allows us to enumerate all the implicit local n-ary constraints, or find the minimal equivalent networks (Montanari 1977), without doing a global search.

Acknowledgments

The work reported here is partially funded by General Electric Aerospace Business Group. I would like to thank Joseph Mundy, Deepak Kapur, and David Cyrluk for many discussions, and Tomas Lozano-Perez for pointing me to papers on network consistency.

Bibliography

Clowes, M.B. (1971) "On Seeing Things," *Artificial Intelligence,* Vol. 2, No. l,pp. 79-116.

Freuder, E.C. (1978) "Synthesizing Constraint Expressions," *Communications of the ACM,* Vol. 21, No. 11.

Henle, M. (1979) *A Combinatorial Introduction to Topology,* Freeman and Co., San Francisco.

Hocking, J.G., G.S. Young (1961) *Topology,* Addison-Wesley Co., Reading.

Huffman, D.A. (1971) "Impossible Objects as Nonsense Sentences," *Machine Intelligence,* Vol. 6, pp. 295-323.

Mackworth, A.K. (1977) "Consistency in Networks of Relations," *Artificial Intelligence,* Vol. 8, pp. 99-118.

Mackworth, A.K., E.C. Freuder (1985) "The Complexity of some Polynomial Network Consistency Algorithms for Constraint Satisfaction Problems," *Artificial Intelligence,* Vol. 25, pp. 65-74.

Malik, J. (1985) "Interpreting Line Drawings of Curved Objects," *PhD Thesis,* Stanford University.

Markowsky, G., M.A. Wesley (1980) "Fleshing out Wire Frames," *IBM Journal of Research and Development,* Vol. 24, No. 5.

Montanari, V. (1976) "Network of Constraints: Fundamental Properties and Applications to Picture Processing," *Information Science,* Vol. 7, pp. 95-132.

Nguyen, V. (1987) "A Parallel Algorithm for Labeling Polyhedral Images," submitted for publication in *Artificial Intelligence.*

Walts, D.L. (1975) "Understanding Line Drawings of Scenes with Shadows," *The Psychology of Computer Vision,* P.H. Winston (ed), McGraw-Hill, New York, pp. 19-91.