

Heuristically Guided Polygon Finding

K. C. Wong, J. Kittler and J. Illingworth
Dept. of Electronic and Electrical Engineering,
University of Surrey,
Guildford, Surrey GU2 5XH, United Kingdom

Abstract

In this paper we present a new algorithm for grouping 2D line segments into open and closed polygons that correspond to feasible physical 3D structures. The algorithm starts by identifying junctions made of two line segments and then forms triples by combining pairs of junctions which share a common line. These triples are then scanned by a procedure which connects them into polygon structures. Heuristic rules are used to control the combinatorial explosion associated with unconstrained associations of junctions and triples. Physical rules are used to reject polygons which are incompatible with a single planar surface hypothesis. The algorithm does not require strict connectivity of end-points at junctions. The polygon finder is seen as a high level grouping step which will enable the recognition of 3D objects which contain flat polygonal faces. Its use is illustrated on real data.

1 Introduction

Recognising 3D objects from 2D geometric data is a difficult problem as explicit depth information is not preserved during the process of projection. However there are important 2D cues which are indicative of specific 3D structures. In particular, an empty area bounded by three or more linear segments, i.e. an open or closed polygon, is most simply interpreted as a flat surface. Thus the recognition of polygonal image curves provides important information which can be used in model-based recognition. In this paper we present a novel algorithm for the detection of polygonal curves. The algorithm exploits specific heuristics which greatly reduce the combinatorial explosion associated with unconstrained combination of linear segments into the higher level polygonal structures. The results of polygon finder will be useful for both object recognition and pose determination.

2 Junctions and triples

The polygon finding algorithm begins with a set of straight line segments extracted from an image. This can be achieved from gray level data by using edge finding followed by a Hough-based line finder [1]. A basic concept in the algorithm is that of a *junction region*. The definition adopted follows the work of Horaud et al[2] on perceptual grouping. A junction region is an image window which contains one or more line segment end-points. The maximum size of a junction region is predefined and line segment comparisons are made to

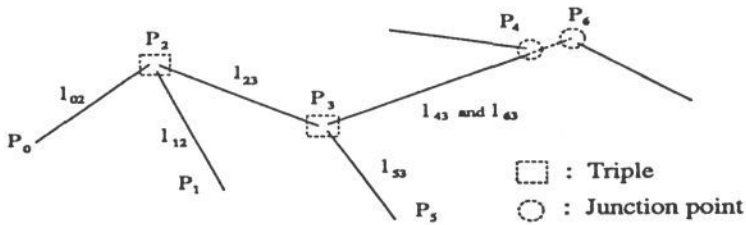


Figure 1: An example of junction configurations

determine whether line segments endpoints are mutually proximal to within this value. Simple geometric tests can be used to limit the comparisons which are necessary to determine this. Each junction region is given a unique integer identifier, R_i , and line segments are then designated as $l_{i,j}$ where the subscripts relate to the junction region labels of their constituent endpoints, see Figure 1.

The next step of the algorithm proceeds by considering merging line-pairs which share a common junction point and thereby constitute a V. Pairs of V-junctions are grouped to form "triple" line structures subject to certain constraints. The use of triples was recently suggested by Henikoff and Shapiro [3] but in our work we refine the concept by introducing a distinction between "weak" and "plausible" V-junction types. A "weak" V-junction is one in which the interior angle subtended by the two lines includes a third line segment. In Figure 1 the V-junction formed by the pair $\{l_{02}, l_{23}\}$ is a weak junction while $\{l_{23}, l_{43}\}$ is a plausible junction. The importance of the distinction is that the plausible junctions are more likely to be representative of significant physical structure and therefore although all V junction types are considered, the plausible junctions are considered first in the search process. This ordering reduces the computational expense of the search process.

In forming triples the two merging V-junctions must have a common line segment and their resultant triple must have only two endpoints. The triple is called a "weak" triple if either of the constituent junctions is a "weak" junction. A "plausible" triple is formed using two "plausible" V-junctions. In many cases there are several V-junctions at each junction region and it is then necessary to choose to combine into a triple only that V-junction whose subsequent continuation is likely to yield a physically significant closed polygon. Arbitrary investigation of all possible continuations is computationally expensive and therefore heuristics are used to limit this process. To appreciate these heuristics consider Figure 2(a) which shows a situation where there are two possible V-junctions $\{l_{k2,k1}, l_{k2,b1}\}$ and $\{l_{k2,k1}, l_{k2,b2}\}$ that might be merged with junction $\{l_{k1,s}, l_{k1,k2}\}$ to form triples. The common line of the triple, $l_{k2,k1}$ is called the linking segment and its extension divides the 2D plane into two half-planes. Line $l_{s,k1}$ is known as the seed segment while the remaining lines are possible branching segments. The choice of V-junction to include in the triple can be decided by considering the angular relationship between the seed segment and the branching segment and whether their endpoints lie in the same or different half planes. Only three distinct situations need be considered. In a case where the two branching endpoints lie in different half-planes then the best branch to

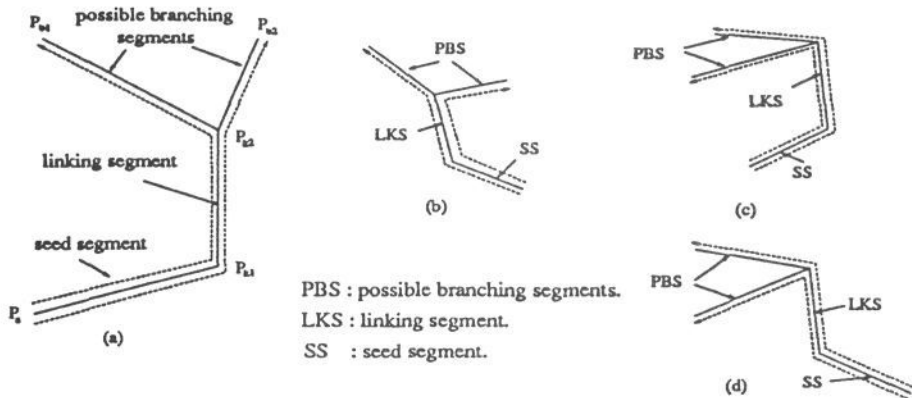


Figure 2: (a) Two possible interpretations of a triple (b) Only one of the two possible branching points is located on the same side of a linking segment (c) Two possible branching points are located on the same side of a linking segment (d) Two possible branching points are located on the different side of a linking segment.

choose is that one whose endpoint is in the same half-plane as the seed segment endpoint, see Figure 2(b). If both branching segment endpoints lie in the same half-plane as the seed segment endpoint then the segment to choose is that one whose dot-product with the linking segment is largest, see Figure 2(c). By contrast, if both branching segment endpoints lie in the half plane opposite the seed segment endpoint then it is desirable to choose the branching segment whose dot-product with the linking segment is smallest, see Figure 2(d).

A further heuristic which is important in constraining the search effort is the identification of "isolated triples". These are triples whose two interior junction regions consist of only the two segments which take part in the triple. Isolated triples are extremely significant and therefore in the merging processes which form triples into polygons they are considered early in the search.

3 Merging to form 2D Polygons.

The result of the processing discussed in the last section is a list of triples which are likely to be part of a closed polygon structure. The next part of the algorithm is concerned with the linking of these triples into higher order structures to form these polygons. Once again specific heuristics are used to guide the search to a successful conclusion. Firstly, those single triples whose two endpoints are in the same junction region are identified as triangles and removed from further consideration. Secondly, quadrilaterals are identified as formed from two overlapping triples which share a common endpoint junction regions. These too are removed from consideration before the search is initiated for five or more sided polygons.

In looking for significant polygon chains, a triple from the top of a sorted triple list is chosen. The algorithm attempts to extend this triple by following a chain of triples around in a loop in a single direction. The initial search direction is called the *forward chaining direction*. If attempts to extend the chain in the

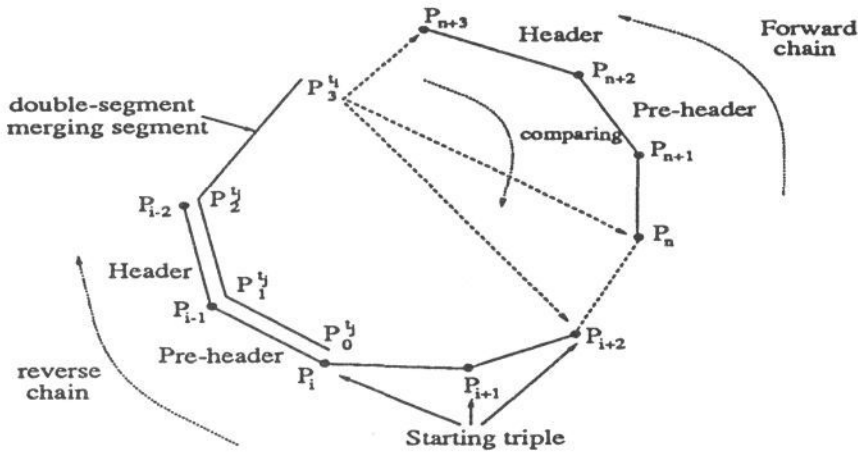


Figure 3: A closed chain test is performed on a processing chain

forward direction fail then the search attempts to extend the chain by starting at the other end and going in the opposite direction. This is known as the *reverse chaining direction*. To help explain the process in detail we introduce some terminology, which is illustrated in Figure 3. The last segment which is part of the chain formed by forward chaining is known as the *forward chain header*. The second last segment of the forward chain is termed the *forward chain pre-header*. The isolated endpoint of the forward chain header segment is called the *forward leading point*. Similar terms apply to the segments and points associated with the reverse chain direction.

When adding a triple to a partial chain three possible situations can be distinguished depending on the overlap between the chain and the triple. These possibilities are known as single point (sp), single segment (ss) and double segment (ds) merges. These are illustrated in Figure 4. In a single point merge the endpoint of the merging triple coincides with the leading point of the chain. In single segment merging the seed or branch segment of the triple and the chain header are the same segment. Finally, a double segment merge occurs when two consecutive segments of the triple are shared with the header and pre-header of the partial chain. At each point of the merging operation there may be for a given partial chain and each merge type several choices for the triple to be appended. For each choice, a closed chain test is first carried out. For example, in Figure 3 when the double segment merging triple is added to the chain the end point P_{n+3}^j is compared with all points preceeding P_{i+2} . If the end point is within the size of a junction region from one of these points then a closed chain is identified. A similar test applies in the case of single segment merging. If there is only one choice then that triple is known as a *unique merging triple*. Unique merging triples are particularly relevant in the formation of significant polygon chains and therefore a heuristic which is used to effectively limit the search process is that we initially attempt to extend the chain using only unique merging triples.

Unique single and double segment merging triples are searched for in the forward direction and if this fails then the search moves in the reverse direction.

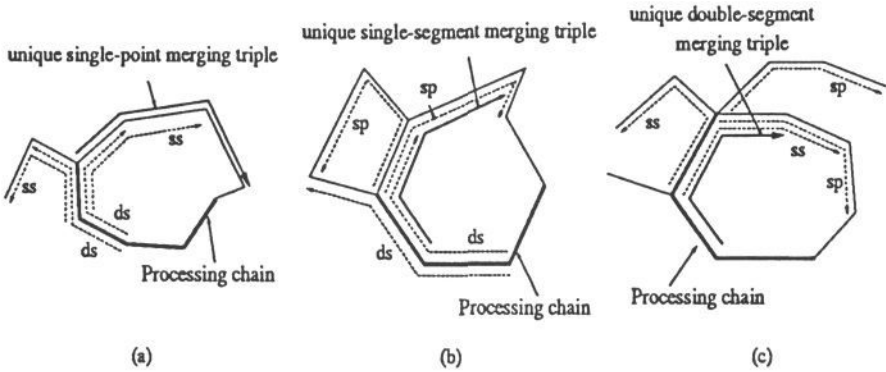


Figure 4: (a) An unique single point merging triple (b) An unique single segment merging triple (c) An unique double segment merging triple.

If a unique merging triple cannot be found in either direction then a further strategy of looking for a *bridging chain* is employed. The bridging operation involves extending the chain at both ends by considering all single point merges and all single segment merges. Double segment merges are not relevant for this bridging process. A closed chain is then identified if the extended chains overlap one another. At any stage of the merging process one of three outcomes may occur:

1. a closed chain may be identified.
2. a chain whose endpoints are isolated may be identified. This is called an open chain.
3. a chain may have potential triples for its extension but none conform with the unique merging or bridging rules discussed above. In this case the chain is said to be unresolved.

In a first pass through the triple data the set of triples which have to be considered in the process is pruned down by removing those which have two consecutive segments shared with an identified plausible closed chain. Triples which are found to be part of open or unresolved polygon chains are not removed from consideration at this stage as they may still be part of other, yet to be discovered, closed chains. A second pass through the reduced data is then done to identify unambiguously the open chains. A feasible open chain should have only one segment shared with a closed chain, hence it is best to use triples which share a middle segment with a segment of a closed chain as a starting triple for initiating the chaining process. An open chain extracted by the grouping process is described by its number of segments and the type of terminating condition. The end point of an open chain can be described by an element of the set $P = \{e, t, u\}$, where e denotes an isolated end point, t denotes the tail of a lambda junction and u denotes an uncertain point where several lines may emanate from the junction. Some examples are illustrated in Figure 5. In Figure 5(b), the darkened chain is described as a feasible open chain having 4

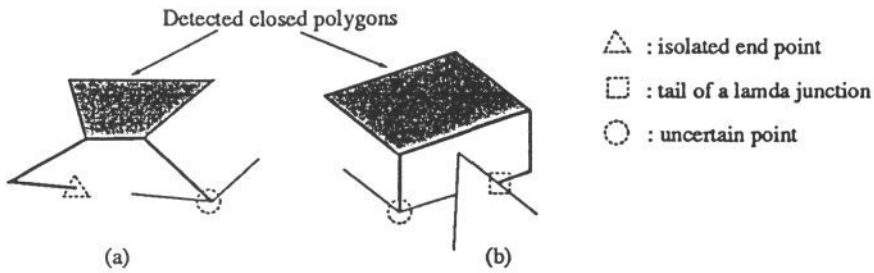


Figure 5: Examples of feasible open polygons

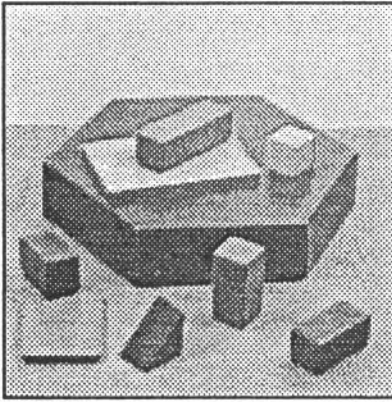
segments and its ends are terminated at an uncertain point and at the tail of a lambda junction. Throughout the triple formation and merging processes the rules of combination have been formulated so that the list of candidate triples is swiftly pruned and hence the computational cost is kept low.

4 Selection of physically feasible polygons

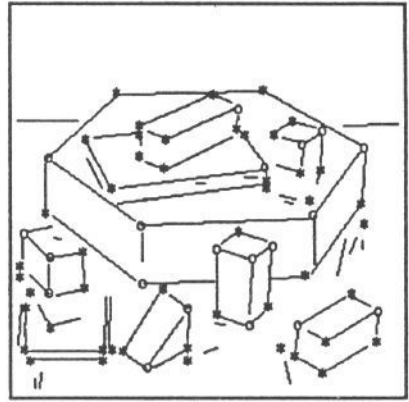
As stated in the introduction the detection of polygons is motivated by the desire to interpret the enclosed regions as single planar surfaces. The majority of 2D polygons detected by the algorithm discussed in the last section represent such physically feasible hypotheses. However some cases correspond to situations which are not physically feasible as single planes. Fortunately these can be rejected by incorporating further more specific, physically based rules into the polygon construction algorithm. Firstly, any 3D edge of a polyhedral object can be shared by at most two different planes. This if a segment of an evolving polygon chain incorporates a segment which has already been identified with two polygons then this line of search can be immediately abandoned. A second physical rule which is extremely useful is that the faces on opposite sides of a segment cannot lie in the same physical plane. Thus it is not possible for a partially developed polygon chain to have two consecutive segments shared with an already identified closed chain.

5 Results

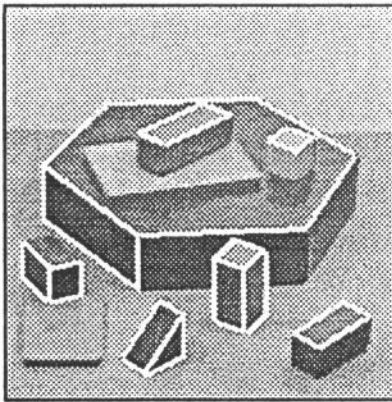
In this section we show some results on real data to illustrate the performance of the polygon finder. The images were taken with a standard CCD camera and were processed on Sun-4 Sparcstation with code written in C. The input images were processed by a Canny edge detector and lines were identified in its thresholded output using a Hough-based line algorithm. Figure 6 (a) and (b) show the gray-level image and subsequent lines extracted by the Hough process, respectively. It can be seen that edges do not always meet exactly at a junction. Figure 6 (b) shows the junctions used in the grouping process, where * is a V-junction and o is a Y or W-junction point. It is important to note that there are many lines in the image which take part in junction formation but do not correspond to physically significant structures. These extraneous



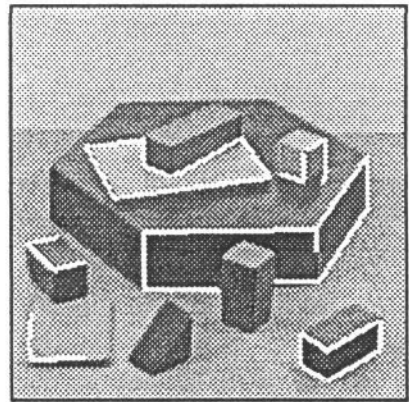
(a) A gray-level image.



(b) Identified lines and junctions.



(c) The feasible closed polygons.



(d) The feasible and infeasible chains.

Figure 6: (a) A gray-level image of a polyhedral scene. (b) A line drawing extracted by Hough process and junctions used in the grouping process. (c) The feasible closed polygons generated from the grouping process. (d) Examples of both the feasible and infeasible polygons extracted during the grouping process

lines are due to effects such as shadowing. Also there is a lot of occlusion in the image caused by the juxtaposition of objects and self-occlusion. Thus the image is a difficult test for subsequent image interpretation.

In this example 48 V-junction were found and 20 Y- and W- junctions were identified. Following triple formation there were 111 triples. Figure 6(c) shows some of the polygons identified by the polygon finding process. Most of the structures in the image are quadrilaterals but the algorithm does successfully identify even the hexagonal shaped box. The program may identify, due to self and interobject occlusions, many sided polygons which are not physically interpretable as being due to single planar faces. However for this image only 2 such infeasible closed polygons extracted from the grouping process. In the case of open polygons, there are 15 open chains generated in the searching process. Some of these examples are shown in Figure 6(d). There are 6 open chains identified which are not part of a real object structure and there are 9 open chains which are part of a real object surface. After the generation of the V-junctions was completed, the generation of the triples, the searching for both closed and open polygons and the selection of the feasible polygons took 1.1 seconds. All single polygonal faces of the objects are found if the initial segmentation processes have correctly located lines that take part in junction formation.

6 Conclusions

In this paper a novel polygon finding algorithm has been presented which works on real data and copes with many of the typical problems encountered in poorly segmented images. Both heuristic and physical rules have been utilised to control the search effort required in the polygon formation process. Experiments are demonstrated on real data and all polygons which could reasonably be due to single planar surfaces are found. Only a few spurious open polygon chains are identified. In future work these planar hypotheses will be used as an index into a database of models for objects which contain at least one planar surface patch.

Acknowledgements

The authors would like to thank J. Matas for providing the software for junction finding and Issac Ng for providing the postscript programs for incorporating the images into the text.

References

- [1] Princen J., Illingworth J., Kittler J., "A Hierarchical Approach to Line Extraction Based on the Hough Transform.", *Computer Vision, Graphics and Image Processing*, Vol 52, pp 57-77. (1990).
- [2] Horaud R. and Veillon F., "Finding Geometric and Relational Structure in an Image", 1st ECCV, pp 374-384, Antibes, France, 1990.
- [3] Henikoff J. and Shapiro L.G., "Interesting Patterns for Model-Based Machine Vision", 3rd ICCV, pp 535-538, Osaka, Japan, 1990.