Optimal Grouping of Line Segments into Convex Sets[1]

B. Parvin and S. Viswanathan
Imaging and Distributed Computing Group
Information and Computing Sciences Division
Lawrence Berkeley National Laboratory, University of California
Berkeley, CA 94720

ABSTRACT

In this paper, we present a technique for grouping line segments into convex sets, where the line segments are obtained by linking edges obtained from the Canny edge detector. The novelty of the approach is twofold: first we define an efficient approach for testing the *global* convexity criterion, and second, we develop an optimal search based on dynamic programming for grouping the line segments into convex sets. We show results on real images, and present a specific domain where this type of grouping can be directly applied.

1 Introduction

Perceptual grouping has been an active area of research in the computer vision community [1, 2, 5, 9, 4, 11], and some researchers view it as an integral part of any high level reasoning or object recognition task. A typical application for grouping is object detection. In general, object detection by any local process is ambiguous. The ambiguities emanate from noise and changes in contrast–introduced by the low-light-level imaging– and the lack of global feedback inherent in the local pixel processing. In addition, most techniques in low level processing assume a certain model for the underlying local pixel distribution; it is only an approximation and does not hold at all times.

This paper deals with a particular type of grouping that involves searching for convex objects [6, 7]. In this context, it is believed that convexity

[1]This work is supported by the Director, Office of Energy Research, Office of Computation and Technology Research, Mathematical, Information, and Computational Sciences Division, of the U. S. Department of Energy under contract No. DE-AC03-76SF00098 with the University of California. This document is LBL-37489.

BMVC 1995 doi:10.5244/C.9.63

is a significant perceptual cue since it remains invariant under perspective transformation. Furthermore, a number of object recognition systems have relied on finding convex groups that correspond to sub-parts [2]. However, we suggest that grouping based on convexity is only one step of the computational process in mid-level vision, and additional constraints, in the form of high level filters, such as symmetry, contrast, and color are important aspects of any interpretation system. Our system extract edges with the Canny edge detector [3], constructs line segments using iterative line fitting to linked edges, and groups line segments into convex sets. The main novelty of our system is twofold. First, we define a new notion of global convexity that is simple to compute, and second, we define a search strategy that is globally optimum and is based on dynamic programming.

In the next section, we briefly review the past work, and then outline the details of our technique. Finally, we present the result of the grouping process, address its limitations, and point out to additional constraints that are needed per specific domain.

2 Past work

The first work on grouping of isolated line segments into a convex set is due to Huttenlocher and Wayner [6]. The main novelty of their system is in the scale space invariant representation of the local neighborhood function that is based on constrained triangulation. The technique has a time complexity of $O(nlog(n))$ that is dominated by the triangulation. The convexity test is local and grouping for line segments is essentially a greedy based technique. Jacob [7] also developed a technique for grouping sparse line segments into convex sets, which is based on local convexity test and back-tracking search strategy. The main difference between our work and previous research is two fold. First, we propose an efficient *global* convexity test, and second, we develop an optimal grouping strategy that is based on dynamic programming. One immediate result of the global convexity test is that the spiral effects [6] can be eliminated altogether. Second, global optimization enhances the noise immunity of the grouping process. Finally, we also believe that our approach has a simpler underlying structure than the previous research in this area.

3 Description of method

In this section we summarize different computational steps in the the convexity grouping process. The edge detection is based on Canny's approach [3], which is inherently a gradient operator. The resulting edges are linked, curve segments are extracted, and polygon representations of these curve segments are obtained. In general, due to noise and variation in contrast, the edge detection technique produces broken and undesirable curve segments. The objective is to group these curve segments such that individual objects can be extracted from background. The local neighborhood is established by constructing a list of *candidate* line segments that lie within a distance G_{thresh} from an end point of a line segment. This distance is selected empirically, and it is one of the system's parameters. The candidate list provides a set of potential hypotheses for grouping line segments into convex sets.

3.1 Convexity grouping

We envision that each line segment corresponds to a node in a disconnected attributed graph, and the goal of the grouping is to link the nodes in this sparse graph in such a way that convex sets are manifested. In this context, the grouping problem is a function of two entities:

$$Objects = Group(features, geometric\ constraints) \quad (1)$$

In this formulation, *features* correspond to line segments (nodes) and attributes such as length, position, and direction. And the *geometric constraints* represent the relationship between the nodes of a convex object as described by line segments. The goal of the convexity grouping is to link these mid level features, represented as nodes of a disconnected graph, in such a way that accumulation of these nodes remains consistent with respect to the geometric constraints.

The geometric constraints are expressed in terms of the relationship between neighboring line segments. Let S be a convex set that consists of ordered line segments A_1, A_2, .., A_k, i.e., A_1 and A_k are the first and last line segments respectively, as shown in figure 1. The convexity test for adding segment X to S is as follows:

1. let C be the line segment connecting line A_k to A_1,
2. let D be the extension of the line A_k,
3. let α be the angle between line segments X and D,
4. let β be the angle between line segments X and C,
5. let ϕ be the angle formed between line segment D and C,
6. then line segment X can be appended to set S to form a new convex set if the following two conditions are satisfied:

(a) $\alpha + \beta \approx \phi$, and

(b) segment X does not intersect segment A_1.

The importance of our test is that only the first and last line segments of a set are necessary and sufficient for convexity verification. As a result, efficient implementation is feasible.

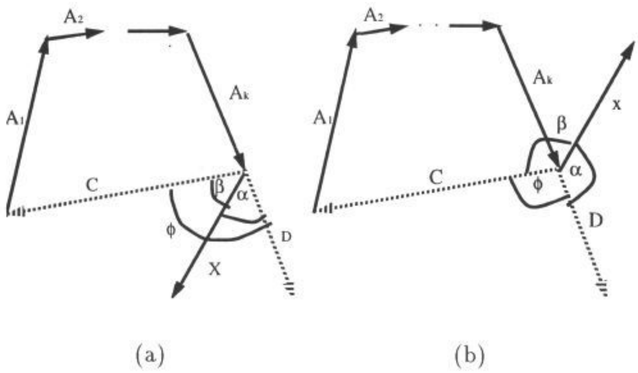


Figure 1: (a) A_1, A_2, A_3, and X form a convex set; (b) A_1, A_2, A_3, and X form a concave set.

The grouping algorithm is initiated by selecting a seed line segment as the initial hypothesis. In our implementation, the seed segments are ranked against their length for generating convex objects. Once the seed is selected, it is used to prune a path for computing a convex set in the direction of line segments, where the directions of the line segments are dictated by the Canny edge detector. However, the seed segment might be in the middle of a convex set. Hence, once the last segment in the convex set is identified, it is used as a seed segment and a backward search for finding a new convex set is initiated. It is possible that using this strategy, some of the line segments that were included in the forward grouping process may not be included in the reverse direction. Nevertheless, forward and reverse search are necessary to capture all the line segments that belong to a given set, and to ensure optimality.

The technique for finding an optimal path for a convex object is based on dynamic programming [10]. This is achieved by defining a cost function where desirable properties are directly encoded. Let

1. L_i and L_j be the length of two adjacent line segments A_i and A_j,

2. g_{ij} be the gap size between line segments A_i and A_j, where the gap size is measured from the proper end points,

3. α_{ij} be the angle between line segments A_i and A_j,
4. S be the current convex set at iteration i (only the first and last line segments are needed),
5. $\Gamma(S, A_j)$ be a binary constraint of 1 or $-\infty$ that tests the convexity hypothesis of adding segment A_j to S.

We define the local cost function between segments A_i and A_j to be

$$cost_{ij} = \begin{cases} \Gamma(S, A_j)\frac{L_i * L_j}{g_{ij}} cos(\frac{\alpha_{ij}}{2}) & \text{if } g_{ij} < G_{thresh} \\ -\infty & \text{otherwise} \end{cases} \quad (2)$$

The above cost function favors grouping those line segments that generate long line segments, with small gaps between them, while maintaining some degree of collinearity in the group. This cost function is then integrated over the entire path of a convex set, and the path with maximum cost is then selected for a given seed segment. In this fashion, the path that satisfies closure, convexity, and optimality is extracted. The dynamic programming algorithm is essentially a multi-stage optimization technique where at each stage, or each iteration, the size of the path is increased by one line segment, and the cost of that particular path from the initial seed segment to the last line segment is propagated. This process continues until no more line segments can be added to the list from a given seed point.

3.2 Optimization

Dynamic programming is a method for solving sequential decision problems [10]. Let P be a set of states, D be a set of possible decisions, $F : P \times D \mapsto \mathcal{F}$ be a cost function, and $\psi : P \times D \mapsto P$ be a function that maps the current state and a decision into the next state. In a single step, the maximum possible value starting from state p_i is given by:

$$H_1(p_i) = \max_{d \in D} F(p_i, d) \quad (3)$$

By the same token, choosing a decision d that maximizes the value of a sequence for n states starting from p_i is found by:

$$H_n(p_i) = \max_{d \in D} [F(p_i, d) + H_{n-1}(\psi(p_i, d))] \quad (4)$$

The above recurrence relation, together with the cost function of equation (2), specifies an optimum path for the refined contour such that constraints are satisfied. In this formulation, the decision d corresponds to any of the candidate line segments that correspond to A_j in equation (2).

3.3 Examples and conclusion

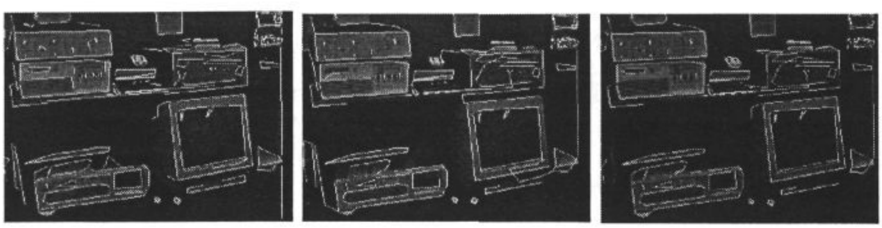
In this section, two examples of the convexity grouping are provided. In these examples, the parameter G_{thresh} was set to 30 pixels. This algorithm was initially developed for detection and tracking of precipitates observed under a transmission electron microscope. It turns out that precipitates have convex geometrical representation that may satisfy other constraints such as parallel or circular symmetries as well. These images are generally noisy, have poor contrast, and depending on the position of the electron beam and the foil angle, suffer from shading artifacts. An example is shown in figure 2, where a, b, and c correspond to the original image with Canny edges overlaid on it, and the result of forward and backward groupings respectively. Notice that the precipitate contains inner structures that are of no significance, since we are interested only in global shape features. In parts b and c, we show the results of forward and backward searches for convexity. It is quite possible that the search could be initiated from a line segment that is not at the start of a sequence, and hence, both forward and backward search is necessary to capture all the line segments that constitute a convex set. Several convex sets are detected, but, only one of them corresponds to the real object. In this case, the desired group has higher contrast and enjoys parallel symmetry. In our system, once an object is extracted, it is tracked with a variant of the snake model [8] for dynamic shape analysis. In this context, the initial contour is coarsely localized by the convexity grouping, and the snake is used for refinement and tracking. The next example is the result of groupings of line segments that correspond to a view of a room. In this example, some of the convex shapes are delineated. And some of the convex sets have no underlying perceptual significance. The latter is due to the fact that convexity is only one intermediate step in the mid-level vision and other constraints such is symmetry, contrast, and color are important cue for any high level interpretation as well.

The optimal convexity grouping algorithm has a time complexity of $O(nm)$ where n and m correspond to the number of line segments and the number of line segments in a given neighborhood respectively.

Acknowledgment: The authors thank Dr. Ulrich Dahmen for providing the data set for this research.

(a) (b) (c)

Figure 2: Search for convex precipitate: (a) Original edges from Canny edge detector; (b) Results of forward search for convexity; (c) Results of forward and backward search for convexity



(a) (b) (c)

Figure 3: Search for convex objects in a room image: (a) Original edges from Canny edge detector; (b) Results of forward search for convexity; (c) Results of forward and backward search for convexity

References

[1] I. Biederman. Recognition by components. *Psychological Review*, 94:115–147, 1987.

[2] R. A. Brooks. Model-based three-dimensional interpretations of two-dimensional images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(2):140–150, 1983.

[3] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986.

[4] S. Grossberg and E. Mingolla. Neural dynamics of surface perception: Boundary webs, illuminan ts, and shape-from-shading. *Computer Vision, Graphics, and Image Processing*, 37:116–165, 1987.

[5] A. Huertas, W. Cole, and R. Nevatia. Detecting runways in complex airport scenes. *Computer Vision, Graphics, and Image Processing*, 51(2):107–145, August 1990.

[6] D. Huttenlocher and P. Wayner. Finding convex groupings in an image. *International Journal of Computer Vision*, 8:7–29, 1992.

[7] D. Jacob. Robust and efficient detection of convex groups. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 770–771, 1993.

[8] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 259–268, London, England, June 1987.

[9] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[10] Bellman R. *Dynamic Programming.* Princeton University Press, 1957.

[11] A. Sha'ashua and S. Ullman. Structural saliency: the detection of globally salient structures using a locally connected network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 321–327, Tampa, FL, 1988.