Controlling the Search for Convex Groups

Francisco J. Estrada and Allan D. Jepson

Department of Computer Science
University of Toronto
10 King's College Road
M5S 3G4, Toronto, On. CANADA
{strider, jepson}@cs.toronto.edu

Technical Report CSRG-482, January 2004

Abstract. This paper describes an efficient algorithm for the perceptual grouping of line segments. The method uses a geometry-based measure of affinity between pairs of lines to guide group formation, and implements a search control procedure that is intended to reduce search complexity when image characteristics lead to a combinatorially large number of possible groups. We also present a ranking system that identifies the polygons that offer the most plausible explanation for the observed image data. The method is applied in the context of finding convex groups, and is experimentally shown to outperform existing algorithms, particularly in images with significant clutter, strong texture, and long, curved contours.

1 Introduction

Perceptual Grouping in real world images is an important problem in computer vision. It deals with the organization of a set of features into groups that are not likely to be accidental, and therefore, indicate the presence of interesting world structure. The identification of feature groups can greatly improve the performance of algorithms that carry out recognition, tracking, and correspondence matching. Grimson [3] has shown analytically that search complexity for recognition algorithms can be reduced from exponential to polynomial if they operate on subsets of features, where the groups groups have the property that most of their elements are likely to come from a single object. Clemens and Jacobs [2] show that in the context of object recognition with a large database of models, indexing must be preceded by perceptual grouping to become practical.

Our goal here is to develop an efficient framework for generating perceptually salient feature groups in real-world images with significant clutter and rich texture. The framework should identify groups that are highly likely to correspond to a single object, and be efficient regardless of scene complexity. Our method has been developed in the context of finding closed, convex groups of line segments. It will be shown that our algorithm provides a significant advantage over existing methods, particularly when the number of groups that can be generated in an image is very large. We present an efficient way to guide group formation,

address the problem of keeping the search manageable, and explore the selection of the most plausible groups out of a possibly large set of convex polygons.

2 Previous Work

There has been much research in the area of perceptual grouping. Lowe first applied perceptual grouping to the task of object recognition [9], [10]. His system used properties such as proximity, collinearity and parallelism to generate candidate groups for matching against known object models. Ullman and Sha'ashua [21] propose a locally connected network for curve extraction that determines saliency using smoothness, continuity, and curve length. Mohan and Nevatia use geometric relationships such as proximity, co-curvilinearity, symmetry, and continuity to group edgels into a description hierarchy [13], [14]. Their hierarchy goes from edgels and line segments to patches that correspond to parts of visible surfaces. Sarkar and Boyer [19] introduce a voting based scheme for grouping that uses Bayesian Networks to infer structure from subsets of features. Voting and inference are performed at multiple levels, generating a grouping hierarchy. Guy and Medioni [5] propose an algorithm based on a tensor voting scheme with communication between neighboring features, and subject to constraints such as co-surfacity and good continuity. In the particular case of 2D line segments, their method allows for the identification of curve segments and junctions [4], and is able to perform perceptual completion of smooth image contours from fragmented data.

The use of an affinity measure was motivated by work in the field of spectral clustering. Ng, et. al. [15] demonstrate the use of pairwise affinities for data clustering, Shi and Malik [22] describe the use of spectral methods for image segmentation, and Malik, et. al. [12] show how texture and contour information can be incorporated to enhance segmentation results. Perona and Freeman [17] propose that an approximation to the pairwise affinity between scene elements can be used to estimate feature saliency, and use it to separate foreground and background elements. More recently, Mahamud, et. al. [11] propose a contour extraction algorithm that uses properties of the eigenvalues of an affinity matrix to detect salient edges and links. The affinity values are related to the random walk probabilities of particles going from one edge to another, and incorporate proximity and smooth continuation. Salient contours are identified as connected components on the estimated link saliencies, and the procedure is repeated to obtain a pre-determined number of contours.

Contour extraction is also addressed by by Elder and Zucker [1]. Their approach uses a sparse graph whose edges are weighted by the likelihood of two image segments belonging together, and performs contour extraction by searching for minimal-weight cycles within the graph. Ren and Malik [18] describe a multi-scale contour extraction algorithm based on higher-order Markov Models, the model parameters are learned from a database of human segmented images. Saund [20] proposes a search-based algorithm that uses pre-computed preferences for particular configurations of consecutive edges in a contour. These

preferences incorporate proximity and domain specific constraints to keep the problem tractable. The algorithm is shown to efficiently extract contours from sketches and line drawings, but is not designed to deal with fuzzy contours, or richly textured scenes. The identification of convex groups has been addressed by Huttenlocher [6], and Jacobs [7] among others. Jacobs' work is particularly relevant to our discussion, and will be addressed further in the next section. Pao. et.al. [16] discuss the importance of convexity in human perception, and present a convexity-based model for figure-ground separation that is biased toward compact shapes.

Even though previous methods have proven to be successful for analyzing images with particular constraints, real world images with significant clutter, rich texture, and highly organized structure continue to be challenging for perceptual grouping algorithms. As image complexity grows, so do the number and size of groups that can be generated. It also becomes increasingly difficult to distinguish good groups from accidental clusters of features.

3 Convexity and Coverage

Convexity is a powerful perceptual grouping principle. Jacobs argues in [7] that convex chains of edges are unlikely to be accidental. This means that any convex groups that are found in an image are likely to have a common origin. Convexity is also useful since many objects are either convex, or made of convex parts. With this in mind, [7] presents an algorithm for efficiently locating convex groups in a set of lines. A coverage measure, which is the ratio of the part of the polygon that is covered by image edges to its total perimeter, is used to prune potential groups that fail to meet a user-defined threshold. It is shown in [7] that the coverage based algorithm will be efficient if the number of polygons extracted from the image can be kept small by choosing an appropriate threshold.

However, in many real world images the number of convex groups can be quite large. Figure 1 illustrates the problem. There are two reasons for the large number of convex polygons in these images: a) there are combinatorially many ways to groups texture and boundary edges to form convex polygons; and b) in images with long, fuzzy contours, many combinations of adjacent edges lead to convex shapes, in fact, the number of groups grows exponentially with the length of the contour. In images such as these, it is not possible to choose a coverage threshold that will find most of the salient groups, and at the same time, keep the search manageable.

4 Using Affinity to Group Lines

To reduce the problems described above, we need to be able to choose the most promising search paths among a possibly large number of alternatives, and we have to find a way to reduce the total number of paths that are examined. To identify promising paths, we have defined a geometry-based affinity measure that

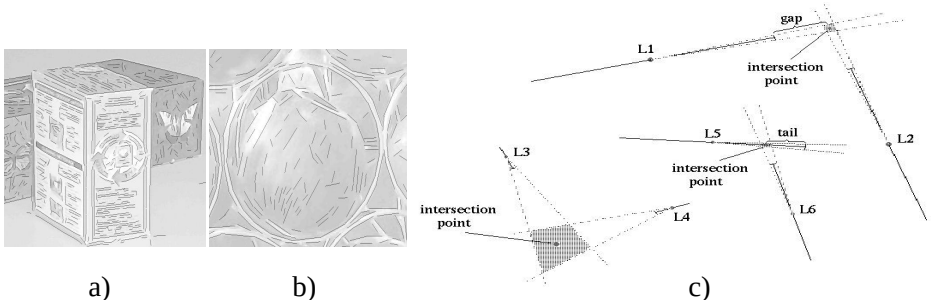

Fig. 1. a) Line-set extracted from a textured box; there are 329,467 convex groups with more than 95% coverage in the line-set. b) Section of a scene where several concentric edge chains are found, there are 4,726,377 convex groups with coverage greater than 95% here. c) Typical image junctions are characterized by an intersection point, gaps, and tails. Due to orientation uncertainty the intersection point can be anywhere within the shaded area. Notice that the uncertainty is greater for smaller segments.

evaluates the quality of the junction formed by two line segments, typical image junctions are illustrated in Figure 1c.

A perfect junction has both segments terminating exactly at the intersection point. If there are gaps between the segments and the intersection, or if segments are split (causing a tail), the junction's affinity decreases. The contribution of gaps is weighted using a Gaussian PDF whose sigma value is related to the size in pixels of the gaps we expect to occur due to imperfections in the line extraction process and regions with low image contrast. We've found a value of $\sigma_{gap} = 20$ to work well in our test images. Tails are also weighted with a Gaussian PDF, but with $\sigma_{tail} = \sigma_{gap}/2$. The reason for the different sigmas is that we expect junctions that originate at the vertices of convex objects to have little or no tails, so we would like the affinity to decrease fast for junctions that cause significant splitting of existing segments. Depending on the geometric configuration, we calculate for each line one of the following distance-affinity factors:

$$DF_{gap} = e^{-d^2/(2\sigma_{gap}^2)}, \quad DF_{tail} = e^{-d^2/(2\sigma_{tail}^2)} \tag{1}$$

where d is the length of the gap or tail.

We expect line-endpoint estimates to be affected by image noise, low contrast, and the spatial resolution of the filter used to detect edges. This leads to uncertainty in the orientation of lines, with smaller lines being more strongly affected by noise. In Figure 1c, uncertainty in orientation is represented as a pair of cones extending from the center of each line toward the intersection point. The cones are such that their width at the line's endpoint is constant for all segments, and corresponds to the expected uncertainty in endpoint localization. The true intersection point can be anywhere within the overlapping region of the two cones. The area of this region depends on the length of each segment, and on their distance from the estimated intersection. Smaller segments, and lines

that are far away from the intersection lead to larger uncertainty. The second component of our affinity measure accounts for this effect, an uncertainty factor is calculated for each line as $UF = e^{-w^2/(k_u^2)}$ where w is the width of the cone at the intersection point, and k_u is a constant that determines how fast the uncertainty term becomes small as w increases. The larger the value of k_u the weaker the effect of uncertainty on the total affinity. We use $k_u = 15$, so the uncertainty term does not become too small unless w is reasonably big.

We combine the distance affinity factors and the uncertainty factors into a geometric affinity measure given by

$$G\_\text{affinity} = (DF_{line1} * UF_{line1} * DF_{line2} * UF_{line2}), \tag{2}$$

and calculate the total affinity for a pair of segments with

$$T\_\text{affinity} = G\_\text{affinity} + \kappa. \tag{3}$$

Where κ is a suitable constant and constitutes a lower bound for the total affinity. Pairs of lines whose geometric affinity is very small will end up with a total affinity near κ, and will be considered to be equally bad for grouping purposes. The value for κ was experimentally set at $\kappa = .25$. Notice that given the above discussion, a perfect junction would receive a geometric affinity very close to 1, and a total affinity close to 1.25. It is important to note that this affinity measure has no angle component, this enables our algorithm to extract polygons with sharp corners, as well as smooth convex contours.The values for the parameters described above are not arbitrary, it will be argued below that they lead to an affinity function with sufficient discriminative power.

Once the affinity between each pair of lines has been calculated, it can be used for grouping. However, the absolute value of the affinity is not very informative since it depends on many factors that can change significantly from image to image, or even within the same scene. This means that a threshold on absolute affinity would be of little use. Instead, we base our grouping constraint on the following reasoning: The large majority of groups that any given line can form will have very low geometric affinity, they correspond to all the accidental intersections between the segment and other lines all over the image. Instead of applying a threshold on the affinity value, let us choose the best K junctions that can be formed with a particular segment, and normalize their affinities so that they add up to 1. If all of the possible groups were equally good, or equally bad, their normalized affinities would be close to $1/K$, but for most edges there will be some good groups with affinities significantly larger than $1/K$, and many groups whose affinity is close to, or under $1/K$. The latter case indicates dubious or downright bad grouping choices.

To gain insight about the advantages of performing affinity normalization, it is useful to examine the shape of the distribution of normalized affinities. Figure 2a shows the histogram of normalized affinities for one of our images with $K = 20$. Notice that most of the histogram's mass and the histogram's peak are found below the $1/K$ value, and that the distribution has a long tail beyond $1/K$. The shape of the distribution is not accidental, Figure 2b shows

the result of averaging the normalized affinity histograms of 20 real-world images from different domains. Even though the images vary greatly in size, number of lines, and complexity, their normalized affinity histograms have the same shape. From the shape of this distribution it becomes clear that a threshold on normalized affinity would have significant pruning power. Furthermore, the threshold is constrained to be larger than $1/K$ for the reasons described above, and enough of the histogram's mass should be above the threshold value to provide the search algorithm with enough junctions to perform grouping. The range of possible values for the threshold is small enough that a good value can be found experimentally, and once this value has been identified, it can be expected to work well regardless of the image. This last claim is well supported by our experiments.

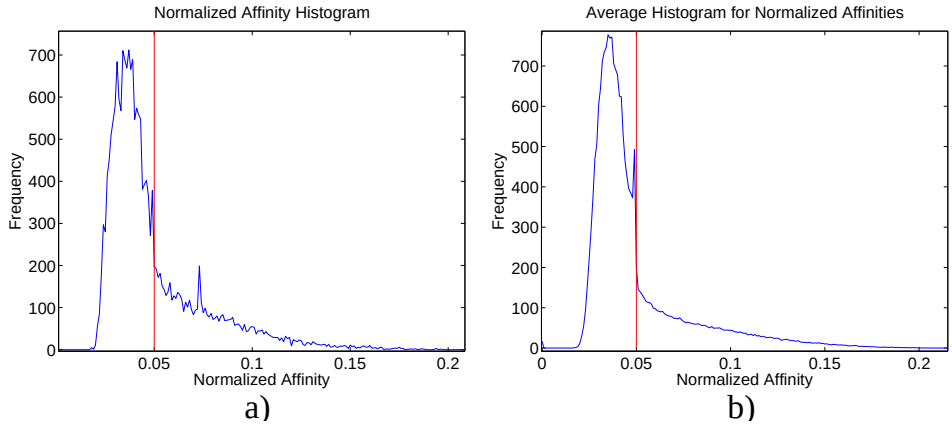

Fig. 2. a) Typical normalized affinity histogram, the red line corresponds to the $1/K$ value. b) Average histogram of normalized affinities for 20 test images. For both histograms $K = 20$

The normalized affinity histograms also provide some information with regard to the discriminative power of our affinity function, Figure 3 shows a normalized affinity histogram for the same image used to generate Figure 2a. We used the same affinity function described above, but we changed the values of the parameters so that $\sigma_{gap} = \sigma_{tail} = k_u = 200$. Under these conditions the difference in absolute affinity between good and bad junctions is minuscule. The resulting histogram has none of the properties discussed above, and indicates that as far as the affinity function is concerned, all junctions look almost equally good.

The particular values we set for the parameters of our affinity function were selected experimentally to be those that provided the best separation between good and bad junctions. The resulting histograms validate the claim that the affinity function does indeed have sufficient discriminative power. The previous discussion also indicates that any affinity function with sufficient discriminative

power will yield a normalized affinity histogram similar to those shown in Figure 2. This is important because it suggests that an efficient search procedure working on normalized affinities could be applied to other problem domains as long as a suitable affinity function can be defined.

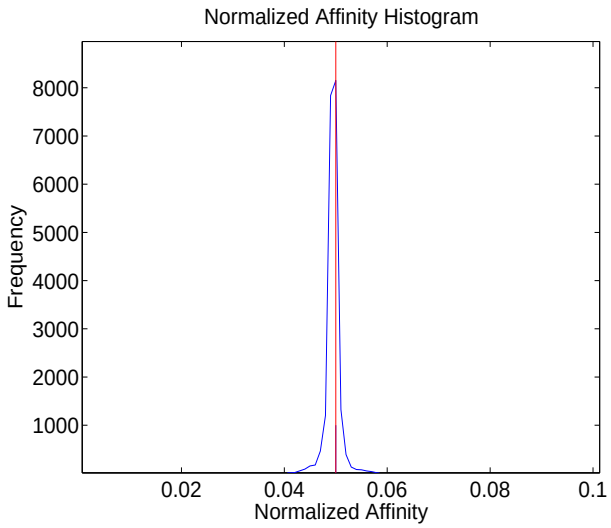

Fig. 3. Normalized affinity histogram for the same image used in Figure 2, but with different parameters for the affinity function. The red line corresponds to the $1/K$ value, and $K = 20$. This distribution lacks the properties described in the text, and indicates that the affinity function has little discriminative power.

Given the above considerations, we propose the following depth-first search strategy:

a) The search procedure will look at each segment in the line-set sequentially, and use it as an initial group.

b) At each step of the search, the algorithm will attempt to generate a larger group by adding an edge the current group, the edge is taken from the list of the best K junctions that can be formed with the last edge of the current polygon.

c) An edge can only be added to the current polygon if its normalized affinity with the last edge of the group is greater than the normalized affinity threshold τ, and if the resulting polygon is convex.

d) The search backtracks when a closed group is found, or when there are no more edges to try in the list of the best K junctions for the last edge in the polygon. A polygon is closed when the last edge and the first edge intersect, and the normalized affinity between them is greater than τ.

We have found that values of $1.3/K \leq \tau \leq 1.5/K$ yield the best results in terms of search efficiency and reliable extraction of convex groups. Such a

threshold allows the algorithm to examine the more interesting grouping choices at each level of the search, while rejecting junctions of dubious quality. The choice of K is not critical, and is domain independent. For small values of K the algorithm becomes increasingly greedy (for $K = 2$, only the best group for each segment is ever examined). For $15 <= K <= 30$ the algorithm performs well. Larger values of K are not required, since we don't expect any segment to be part of more than a small number of significant groups. This procedure achieves a significant reduction in the total amount of search that has to be carried out to detect salient groups. Affinity normalization has the additional benefit of making this strategy robust to variations within the line-set. Locally dense, and locally sparse regions of the line-set are given an equal opportunity to form groups.

However, even though the above grouping procedure greatly reduces the branching factor of the search algorithm, it is not sufficient for dealing with the complexity of scenes such as the one depicted in Figure 1b. In such images, even a small branching factor will lead to a huge search space due to the generation of combinatorially many variations of a few contours.

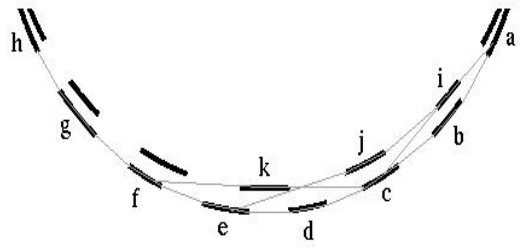

Fig. 4. The figure shows possible chains that can be generated from a particular initial segment. Assume we have found chain a-b-c-d-e-f-g-h, and it leads to a complete polygon. We'd like to prune variations such as a-i-c-d..., a-i-j-e..., a-b-c-k-f..., and so on, since these chains lead to a similar polygon, and the number of variations is exponential on the number of edges that make up the adjacent contours

5 Managing Search Complexity

We would like to avoid searching through chains of edges that are variations of previously extracted polygons (see Figure 4). However, keeping a complete list of distinct groups, and checking each new chain against it would be computationally expensive. Instead, we propose the following algorithm: For each choice of initial segment, initialize an empty list for subsequently extracted polygons. Every time a new group is found, it is checked against the polygons already in the list, and inserted if there are no similar groups already there. If a similar shape is found, we keep the best polygon and discard the other. Section 6 will describe how

to determine which polygon is better, as well as the criteria used to detect equivalence.

To avoid searching through variations of these polygons, we check every new edge added to a chain against the intermediate polygon list. If the edge is covered, we prune the chain since we would expect it to lead to some variation of a polygon that has been found before. This is possible because of the way affinities work. Edges that are very close to each other and have similar orientation will have similar affinities with regard to the surrounding lines, and will produce similar search paths. This procedure ensures that a chain must introduce and subsequently maintain a difference with regard to previously found polygons to be explored, and eliminates the need to search a large space of similar chains that share the same initial segment. However, it does prune out some chains that lead to groups that are distinct, but share part of their contour with previously found polygons. In practice, we expect these groups to be found when the search is started at one of the edges that differentiate them from polygons in the intermediate list. In addition to the above, we've found that search on deep, bushy trees (which occur often along curved, fuzzy contours) can be greatly simplified by increasing the normalized affinity threshold by a small amount at each step of the search (in all our experiments, we used an increment of 1×10^{-6}). The result is that after the search tree has grown considerably, the search is unlikely to proceed unless a really good group is found that justifies the additional effort. Since groups are explored in a best-first order, we expect most salient groups to be found before the search tree has become too large.

Even with the above enhancements, it is not uncommon to find hundreds, or even thousands of groups in a given image. Most of these polygons are variations of a small number of perceptually significant shapes (found by starting the search at different edges along the same contours), the remaining ones are groups that result from accidental relationships between segments. The following section describes a method for determining which polygons are most likely to correspond to interesting structure, and describes our criteria for eliminating most of the similar groups while keeping the best ones.

6 Finding the Good Polygon

To identify the most promising groups, we use the Qualitative Probabilities framework described by Jepson and Mann in [8]. Qualitative Probabilities estimate the log-unnormalized posterior of a model given the image data

$$\log q(M|I) = \log\left(p(I|M)p(M)\right), \tag{4}$$

where $p(M)$ is the prior probability of finding a particular model, and $p(I|M)$ is the likelihood of the image given that model. Jepson and Mann show that the prior probabilities for lines, and objects composed of lines can be specified in terms of a small constant $\epsilon \ll 1$ that gets smaller as image resolution increases. In particular, the probability of finding a line endpoint or polygon vertex at a specified location in the image is of order $O(\epsilon^2)$. Since a line is specified by a

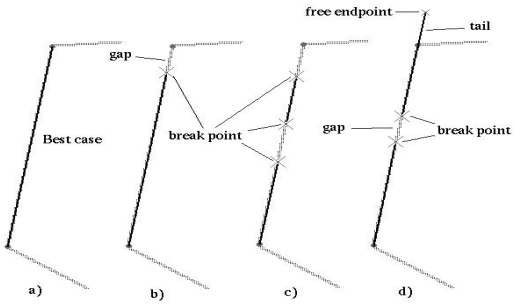

Fig. 5. Fragment of a polygon boundary (gray) and different coverage situations: a) The polygon's vertices explain the location of both line endpoints, no extra parameters are required, so the likelihood is $O(1)$. b) There is one break, its location can be explained with one free parameter (it's distance along the edge), so its likelihood is $O(\epsilon^1)$. c) Three breaks, the likelihood is $O(\epsilon^3)$. d) There are two breaks ($O(\epsilon^2)$), and one tail. The tail is nothing more than a segment which happens to align with the polygon boundary, it requires 3 parameters: 2 for the endpoint we can see, and 1 for the other endpoint which lies somewhere (its exact location is unknown) along the polygon boundary. The second endpoint requires only one parameter since we we know its direction. The combined likelihood in this case is $O(\epsilon^2 * \epsilon^3) = O(\epsilon^5)$.

pair of endpoints, the prior probability of specifying a particular segment is of order $O(\epsilon^2 * \epsilon^2) = O(\epsilon^4)$, and the prior probability of specifying a polygon with t vertices is of order $O(\epsilon^{2*t})$. This corresponds to the term $p(M)$ for a convex group in equation 4. In a similar way, the prior probability of specifying a set of n lines is of order $O(\epsilon^{4*n})$. Under QP, the prior probability for a complete image is given by the multiplication of the prior probabilities for hypothesized groups, and any remaining lines that are not part of these polygons. Since $\epsilon \ll 1$, the larger the order of epsilon (the more endpoints, polygon vertices, and lines that need be specified), the smaller the corresponding probability.

A group increases its posterior probability by decreasing the total number of parameters that are required to account for the observed image data. It does so by offering a cheaper explanation for the lines that compose its boundary. However, in general the match between the model and the observed lines is imperfect, any gaps or tails occurring along the boundary must also be accounted for. The term $p(I|M)$ in equation 4 incorporates these effects. Figure 5 shows part of a polygon's boundary under different conditions of line-set support, and gives a detailed example of how the term $p(I|M)$ is evaluated. The epsilon order of the likelihood term depends on the number of free parameters that must be specified to account for the observed gaps and tails in a particular polygon.

For our grouping method, the QP framework was extended so that the term $p(I|M)$ also accounts for lines that terminate at the boundary of a hypothetical group, as well as lines that are split, and lines that completely cross over the proposed polygon. This allows us to incorporate more evidence into the QP

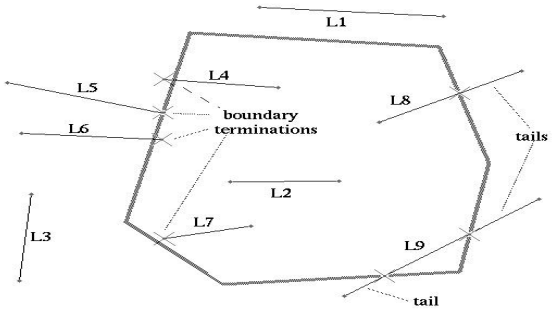

Fig. 6. Typical polygon interactions with the line-set. The polygon is treated as an opaque surface. $L1$, $L2$, and $L3$ are not affected by the polygon, each requires 4 parameters. $L4$ and $L7$ are treated as lines that are 'painted' onto the polygon, and terminate at the boundary (within some tolerance), whereas $L5$ and $L6$ are outside segments obscured by the polygon (the location of their second endpoint is unknown). In both cases, we require 3 parameters to specify the visible part of each line: 2 for the free endpoint, and 1 to specify the location where the segment meets the polygon boundary (as a length along the appropriate polygon edge), hence the likelihood is $O(\epsilon^3)$. $L8$ is split, it is treated as 2 lines terminating at the boundary, one inside, and one outside the polygon, so the split requires 6 parameters and the likelihood is $O(\epsilon^6)$. $L9$ is split in 3 parts, each outside segment is treated as a line obscured by the polygon, and requires 3 parameters. We also need 2 parameters to specify the inside fragment (2 distances, one along each polygon edge), thus $L9$ requires 8 parameters given this particular polygon, and the likelihood is $O(\epsilon^8)$.

calculation. Figure 6 shows several possible situations, and the associated value of $p(I|M)$ in terms of ϵ. Intuitively, $p(I|M)$ is large for a polygon that has few gaps along the boundary, offers an explanation for lines that terminate at its boundary, and does not split segments. The above considerations can be used to efficiently estimate $q(M|I)$ for any polygon. We are now ready to integrate the above discussion into the search framework. Recall that for each initial segment we keep an intermediate list of polygons. At search time, a newly detected group is checked for equivalence against the polygons that are already in the list. Polygons P_a and P_b are said to be equivalent if all the vertices of P_a are within a small distance of an edge in P_b, and vice-versa. If an equivalence is detected, the group with the largest estimated $q(M|I)$ is kept.

Notice, however, that the full evaluation of $p(I|M)$ requires computing line intersections and performing some geometric processing. This would be too expensive if it were to be performed on every polygon that is found, instead, for the intermediate group list we consider in the likelihood term $p(I|M)$ only lines that lie along the polygon's boundary, as shown in Figure 5. This is no more expensive than checking the boundary for coverage. Once all possible distinct groups have been found for a particular initial segment, the contents of the intermediate group list are inserted into a final list of convex polygons, sorted by their posterior probability. This time, however, the complete likelihood term in-

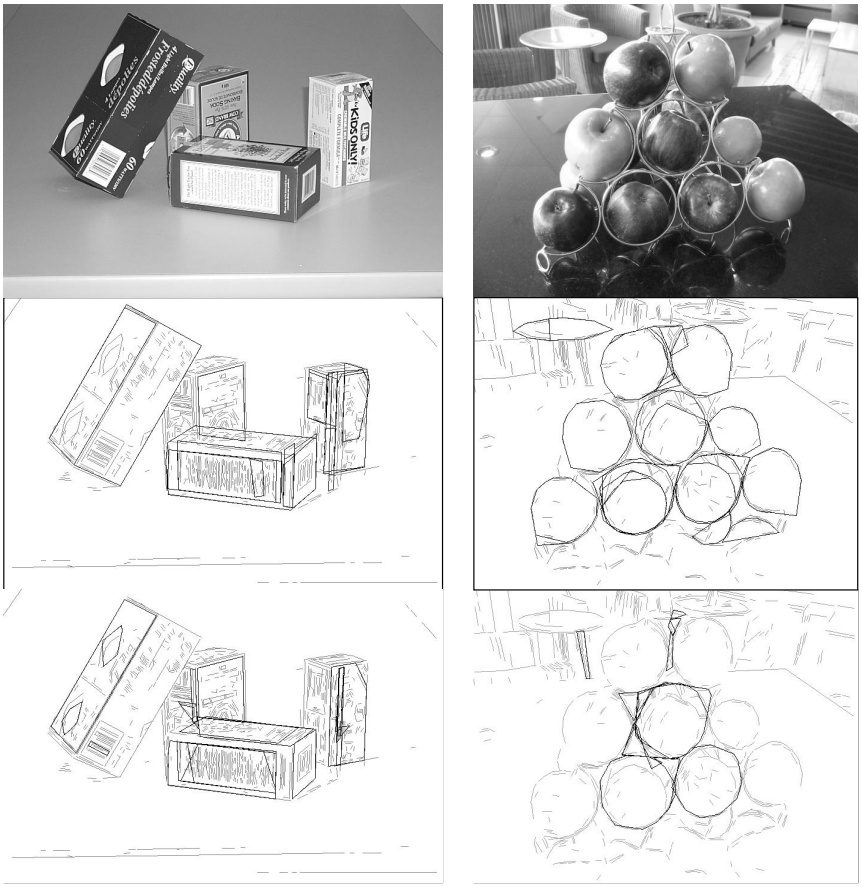

Fig. 7. Test results with $K = 20$ and normalized affinity threshold of $1.5/K$: **a)**The middle image displays the best 20 polygons output by our algorithm, the one on the bottom shows the best 20 polygons found using coverage threshold of .9 and ranking groups based on their coverage fraction. The coverage based algorithm searched through $37,518,725$ nodes, took 58 minutes to complete, and found $89,553$ groups, out of which 90 were distinct. Our algorithm searched through $112,053$ nodes, took 13 seconds to complete, and found 849 polygons out of which 130 were distinct. Most of the polygons displayed in either image (or small variations of them) were found by both algorithms, but the ranking is quite different. The input line-set contains 835 non-oriented segments. **b)**The middle image shows the best 25 polygons output by our algorithm. The bottom image shows *all* the polygons found using a coverage threshold of .95. Even with this very high threshold, the coverage based algorithm searched through 1.99×10^9 nodes, required 42 hours and 42 minutes to finish, and found over 29.7 million groups, yet only 19 of these were distinct. Our method searched through $348,642$ nodes, ran in 56 seconds, and found 852 groups, out of which 78 were distinct. The input line-set consists of 797 non-oriented segments. Run-times were measured on a Pentium 4, 1.9GHz desktop machine.

cluding all interactions between the polygon and the line-set is calculated. Any polygon whose posterior is worse than $O(\epsilon^{4n})$ is discarded since at least as good an explanation for the same image is provided by a model that consists of n independent lines.

The end result of the above procedure is that the final polygon list is significantly smaller than the original number of convex groups detected in the image. Furthermore, the polygons at the top of the list are those that do a better job of explaining the observed line-set and, as will be shown in the experimental results, tend to agree with perceptually salient structure.

7 Experimental Results

This section presents the results of running our algorithm on several images, all tests were carried out using the same parameters as described in Section 4. Figure 7a shows the results of running our algorithm on an image of boxes. Notice that many of the polygons that were determined to be most plausible by the Qualitative Probabilities framework agree with the boundaries of boxes in the original image. Figure 7b shows the results of running our algorithm on the complete version of the fruit image from which Figure 1b was taken. The results on this image provide a particularly clear demonstration of the reduction in search complexity that is achieved by our framework. Notice that although the total number of groups found by our method is much smaller than the number of groups found by the coverage based algorithm, the number of distinct polygons found by our system is much larger.

The experimental results in the fruits image demonstrate the benefits of using the search control procedure described above, the coverage-based algorithm finds over 29 million groups but only 19 describe distinct shapes, this indicates that the search procedure is spending most of its time processing polygons that are only small variations of previously encountered shapes. In contrast, our algorithm finds less than 1 million groups, out of which, 80 are distinct. This is a very significant improvement, in fact, the search control procedure alone yields a reduction of roughly two orders of magnitude in search time for images that contain such a large number of similar groups. Results in the boxes image also show a significant reduction in the number of polygons that were detected by our algorithm, while at the same time, identifying a larger number of distinct shapes. Together, these results indicate that the search control procedure yields a substantial speed-up with a negligible loss in search completeness.

While the benefits obtained from the search control procedure vary depending on the number of similar polygons that can be found in an image, the use of normalized affinities consistently yields a reduction in the number of nodes searched, while at the same time, increasing the number of distinct polygons. This indicates that the affinities, together with the use of QP to rank the polygons, help to focus the search on the more salient groups. Finally, the images selected in our experiments show that the algorithm is able to extract shapes with smooth contours, as well as polygons with sharp corners. The reader is

referred to **http://www.cs.utoronto.ca/~strider/TechRep482/** for more experimental results.

8 Conclusion and Future Work

The above results indicate that our framework yields a very large reduction in the amount of search that has to be carried out in order to identify perceptually salient groups, while at the same time, increasing the number of salient polygons extracted. We have also presented a ranking algorithm that allows us to identify the polygons that do a better job of explaining the observed line-set. In its current form, our method is capable of efficiently processing complex, real-world imagery from different domains.

The geometric definition given here for the inter-line affinities is suitable to the problem of identifying convex polygons, but the affinity measure can be extended to incorporate domain specific constraints, or adapted to work in different domains with features other than line segments. The pruning algorithm based on normalized affinities is applicable to any search problem for which a suitable affinity measure can be defined.

There are several directions for future work. We are evaluating the substitution of convexity with other grouping constraints that allow for the identification of more general shapes, but at the same time, keep the search manageable. We are also looking at the integration of image segmentation results into the above framework, we believe that photometric information can provide additional constraints during the search phase and, once the search is completed, could be used to identify and merge polygons that are likely to be part of the same object.

Photograph credits

The fruits image used in Figure 1b and Figure 7b was kindly provided by Mr. Peter N. Lewis.

References

1. Elder, J. and Zucker, S. "Computing Contour Closure" *ECCV*, Vol. 1, pp. 399-412, 1996.
2. Clemens, D. and Jacobs, D. "Space and Time Bounds on Indexing 3-D Models from 2-D Images" *PAMI*, Vol. 13, no. 10, pp. 1007-1017, 1991.
3. Grimson, W. E. L. "The combinatorics of heuristic search termination for object recognition in cluttered environments" *PAMI*, Vol. 13, no. 9, pp. 920-935, 1991.
4. Guy, G. and Medioni, G. "Inferring Global Perceptual Contours From Local Features" *IJCV*, Vol. 20, no. 1/2, pp. 113-133, 1996.
5. Guy, G. and Medioni, G. "Inference of Surfaces, 3D Curves, and Junctions From Sparse, Noisy, 3D Data" *PAMI*, Vol. 19, no. 11, pp. 1265-1277, 1997.
6. Huttenlocher, D., and Wayner, P. C. "Finding Convex Edge Groupings in an Image" *IJCV*, Vol. 8, no. 1, pp. 7-29, 1992.

7. Jacobs, D. W. "Robust and Efficient Detection of Convex Groups" *PAMI*, Vol. 18, no. 1, pp. 23-37, 1996.

8. Jepson, A., and Mann, R. "Qualitative Probabilities for Image Interpretation" *ICCV*, pp. 1123-1130, 1999.

9. Lowe, D. G. *Perceptual Organization and Visual Recognition*, Kluwer Academic Publishers, 1985.

10. Lowe, D. G. "Three dimensional object recognition from single 2-D images" *AI*, Vol. 31, pp. 355-395, 1987.

11. Mahamud, S., Williams, L., Thornber, K., and Xu, K. "Segmentation of Multiple Salient Closed Contours from Real Images" *PAMI*, Vol. 25, no. 4, pp. 433-444, 2003.

12. Malik, J., Belongie, S., Leung, T., and Shi, J. "Textons, contours and regions: cue integration in image segmentation" *CVPR*, pp. 918-925, 1999.

13. Mohan, R. and Nevatia, R. "Segmentation and Description Based on Perceptual Organization" *CVPR*, pp. 333-341, 1989.

14. Mohan, R. and Nevatia, R. "Perceptual Organization for Scene Segmentation and Description" *PAMI*, Vol. 14, no. 6, pp. 616-634, 1992.

15. Ng, A., Jordan, M., and Weiss, Y. "On Spectral Clustering: Analysis and an algorithm" *Advances in Neural Information Processing Systems 14*, MIT Press, 2002.

16. Pao, H., Geiger, D., and Rubin, N. "Measuring Convexity for Figure/Ground Separation" *ICCV*, pp. 948-955, 1999.

17. Perona, P. and Freeman, W. "A factorization approach to grouping" *ECCV*, Vol. 1, pp. 655-657, 1998.

18. Ren, X. and Malik, J. "A Probabilistic Multi-scale Model for Contour Completion Based on Image Statistics" *ECCV*, Vol. 1, pp. 312-327, 2002.

19. Sarkar, S. and Boyer, K. L. "Integration, Inference, and Management of Spatial Information Using Bayesian Networks: Perceptual Organization" *PAMI*, Vol. 15, no. 3, pp. 256-274, 1993.

20. Saund, E. "Finding Perceptually Closed Paths in Sketches and Drawings" *PAMI*, Vol. 25, no. 4, 2003.

21. Ullman, S. and Sha'ashua, A. "Structural Saliency: The Detection of Globally Salient Structures Using a Locally Connected Network", *ICCV*, pp. 321-327, 1988.

22. Shi, J. and Malik, J. "Normalized Cuts and Image Segmentation" *CVPR*, pp. 731-737, 1997.