Perceptual Grouping for Contour Extraction

Francisco J. Estrada and Allan D. Jepson
Department of Computer Science
University of Toronto, Toronto, ON, Canada
{strider,jepson}@cs.utoronto.ca

Abstract

This paper describes an algorithm that efficiently groups line segments into perceptually salient contours in complex images. A measure of affinity between pairs of lines is used to guide group formation and limit the branching factor of the contour search procedure. The extracted contours are ranked, and presented as a contour hierarchy. Our algorithm is able to extract salient contours in the presence of texture, clutter, and repetitive or ambiguous image structure. We show experimental results on a complex line-set.

1. Introduction

The visual world is an extremely complex environment, the abundance of detail, texture, clutter, and other artifacts complicates significantly the work of algorithms that perform tasks such as recognition, motion analysis, and geometric reconstruction. In order to reduce the complexity of the visual environment to manageable levels, perceptual organization is used to select parts of the image, or subsets of features that are of particular interest for further processing. Contour extraction is one way in which regions of the image that are likely to correspond to objects of interest can be identified. However, the development of a generic procedure that works well on complex images has proven to be extremely difficult. Our goal in this paper is to present an efficient algorithm to carry out contour extraction on comples line-sets extracted from real world images. We present an affinity based method for guiding group formation, discuss an efficient search procedure, and explore the problem of ranking and organizing the extracted contours.

2. Previous Work

There is a significant amount of previous work in Perceptual Grouping and Image Segmentation. Lowe started the systematic use of grouping for object recognition in [8], and [9]. His system used properties such as proximity, collinearity and parallelism to generate candidate groups for matching against known object models. Mohan and Nevatia use geometric relationships such as proximity, co-curvilinearity, symmetry, and continuity to group edgels into a description hierarchy [12], [13]. Sarkar and Boyer [14] introduce a voting based scheme for grouping that uses Bayesian Networks to infer structure from subsets of features. Guy and Medioni propose an algorithm based on tensor voting with communication between neighboring features [4], [3]. It incorporates constraints such as co-surfacity and good continuity, and is capable of performing perceptual completion on fragmented data. Huttenlocher [5], and Jacobs [6] among others have studied the use of convexity as a grouping constraint. Jacobs [6] demonstrated the use of convex groups for indexing in object recognition.

Affinity measures have been used for clustering and segmentation by Ng, et. al. [1], Shi and Malik [16], and Malik, et. al. [11], among others. Their work exploits the properties of the eigenvectors of an affinity matrix generated from image features, and uses these eigenvectors to determine an appropriate segmentation. More recently, Mahamud, et. al. [10] propose a contour extraction method based on the random walk probabilities of particles traveling between edgels in an image. They use an affinity measure that incorporates proximity and smooth continuation, calculate the saliencies of links between image edgels, and extract contours as connected components within the link saliency matrix. Saund [15] presents a search based procedure for contour extraction from sketches. In his algorithm, a table of pre-computed preferences for configurations of consecutive edges is used along with domain-specific constraints to keep the search tractable. The algorithm is able to extract salient contours in sketch art and simple line drawings, but is not designed to deal with fuzzy or textured contours. Even though previous methods have proven to be successful in their particular domains, extracting salient contours from real world images of significant complexity continues to be a challenging problem.

3. Contour Search Procedure

We first introduced inter-line affinities for search control in [2]. It was shown there that on real world imagery, even well constrained search procedures will be confronted with a combinatorially large number of valid groups arising from texture, clutter, and repetitive image structure. It was also shown that in the context of convex group extraction, a search procedure based on a measure of the geometric affinity between pairs of lines yields a very significant reduction in the amount of search that is required to locate salient groups. We will briefly describe the affinity measure and search algorithm presented in [2] and then introduce modifications that make the extraction of general (non-convex) contours practical and computationally efficient.

Our affinity measure is based on the geometric configuration of pairs of line segments in the image. A perfect junction between two edges occurs when both segments terminate exactly at their intersection point. Gaps between the segments and the intersection point, or junctions that split either or both segments are penalized. We will refer to the split part of a line segment as a tail. The effect of gaps and tails on the affinity measure is given by two Gaussian PDF's:

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

where d is the length of the gap or tail, and the σ parameters determine how quickly the appropriate term decreases with increasing d. The values for these constants were set experimentally to $\sigma_{gap} = 20$ and $\sigma_{tail} = \sigma_{gap}/2$. We make the contribution of tails decrease faster since we expect junctions that come from object boundaries to have little or no splitting of line segments. For each line in the junction, DF_{line} is set to one of DF_{gap} or DF_{tail}, as dictated by the particular intersection.

We also incorporated a term that accounts for the small error in line orientation that we can expect due to noise in the line extraction process. Orientation error causes uncertainty about the true location of the intersection between two segments, and depends on the segments' length and the expected error in endpoint localization (usually a couple of pixels perpendicular to the line). The expected uncertainty in endpoint localization w is calculated for each segment, and its contribution to the over-all affinity is computed using another Gaussian PDF: $UF_{line} = e^{-w^2/(2k_u^2)}$ where k_u is a constant that determines how strongly orientation uncertainty affects the inter-line affinity. The value of k_u was set experimentally to $k_u = 8$, so small uncertainties are not heavily penalized.

The above affinity factors are combined into a geometric affinity measure given by

$$G_{affinity} = (DF_{line1} * UF_{line1} * DF_{line2} * UF_{line2}), \;\; (2)$$

and total affinity for a pair of segments is calculated with

$$T_{affinity} = G_{affinity} + \kappa. \qquad (3)$$

Where κ is a suitable constant which constitutes a lower bound on the total affinity, and it set to $\kappa = .25$ (see [2] for details). Our contour extraction procedure is a depth-first search with a single segment as the starting point. The search adds a line at each level subject to geometric constraints until a closed contour is found or a geometric inconsistency has been detected. A great part of the robustness and efficiency of the algorithm comes from the use of normalized affinities, at each step, the best K possible junctions that can be formed with a particular line segment are chosen, and their affinities normalized. A threshold on normalized affinity is used to determine which groups to form, and which to remove from further consideration. The search algorithm also checks for contours that are similar to previously encountered shapes, and keeps the search procedure from exploring combinatorially many small variations of the same contour.

It was shown in [2] that the above search algorithm, coupled to the constraint that extracted contours must be convex, led to an extremely efficient grouping algorithm, capable of finding perceptually salient convex groups in complex imagery. However, the extraction of non-convex contours requires additional effort. If the convexity constraint is removed, the number of possible groups that can be formed in an image with moderate amounts of texture and clutter becomes too large, and renders even our previously efficient search procedure impractical. Hence, other constraints are needed to keep the search manageable. The search algorithm we propose here substitutes compactness for convexity, at each step in the search, the area of the current contour is compared with the area of its convex hull, and the search procedure is only allowed to continue if the ratio of the two is above a threshold. The usefulness of compactness has already been explored by Saund in [15] in the context of contour extraction, but we have found that by itself it is insufficient to keep the search manageable. The problem is that in the absence of other constraints, compact contours can have irregular boundaries that wander through parts of the image where line density is high. In the presence of texture or clutter, the number of possible variations of such contours grows exponentially. To control the exploration and formation of such groups, we use a smoothness term $AF = (\cos(\theta) + 1)/2$, where θ is the angle between the segments. This term multiplies the geometric affinity defined in (2). Whenever there is a clear path for the contour to follow that preserves smooth continuation, the angle term will keep the search procedure from wandering into textured regions of the line-set. At the same time, the affinity normalization procedure ensures that if a sharp turn offers the best grouping choice, its normalized affinity will be above threshold, and the group will be explored. It will be shown in the experimental results that the algorithm is capable of extracting non-convex contours with smooth boundaries as well as sharp corners. The resulting search algorithm is robust, and remains efficient in the presence of texture, clutter, and ambiguous image structure.

4. Contour Ranking and Organization

Output contours are ranked and inserted into a hierarchical structure with the largest, non-equivalent shapes at the top level, and where each contour contains a list of smaller contours found within. To determine equivalence, we compute the percentage of overlap between two shapes, if this percentage is above 95%, we say the shapes are equivalent, and keep only the one with the best ranking. Ranking is based on the Qualitative Probabilities framework described by Jepson and Mann in [7], which estimates the log-unnormalized posterior of a model given the image data

$$\log q(M|I) = \log\left(p(I|M)p(M)\right), \quad (4)$$

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 by imposing weak priors on the probability of occurrence of lines and vertices of polygons, a reasonable estimate of the plausibility of a particular contour can be determined. These weak priors are defined with regard to a small constant $\epsilon \ll 1$, and their magnitudes are shown to be of order $O(\epsilon^2)$ for polygon vertices, and $O(\epsilon^4)$ for isolated line segments.

Consider a closed contour consisting of t line segments, under the QP framework, a polygonal contour with t edges and t vertices, has a prior probability of order $O(\epsilon^{2*t})$, whereas t isolated line segments occur with a prior probability of order $O(\epsilon^{4*t})$. Thus, a contour that contains these t lines offers a more plausible explanation than the accidental occurrence of t independent segments that happen to form a closed loop. Jepson and Mann also describe a method to account for fragmentation along edges of the contour in terms of ϵ. The effects of fragmentation and other imaging artifacts are accumulated onto the $p(I|M)$ in Equation 4, and allow for the estimation of the posterior likelihood of a particular polygon given the observed image data. In [2] we show how the Qualitative Probabilities framework can be extended to incorporate evidence provided by lines that are split by an extracted contour, and lines that terminate at the contour's boundary. Given the above considerations, our ranking method prefers contours that account for a significant number of segments along their boundary, split few or no lines, and preferably run through parts in the image where there is evidence of a separating boundary in the form of lines terminating at the contour.

The complete contour extraction procedure is summarized as follows: a) Starting with each line segment in turn, generate as many closed contours as possible subject to the thresholds on compactness and normalized affinity; b) when a contour is found, calculate its QP ranking; c) check for equivalence against previously extracted shapes, if equivalence is detected keep only the polygon with the highest QP ranking; d) If no equivalence is found, determine the appropriate place for the contour in the contour hierarchy, and insert it. Within the polygon hierarchy, the top level polygons are sorted according to QP rank.

Previous results in [2] demonstrated that in the context of convex group extraction, the QP ranking procedure would robustly identify perceptually salient groups even in the presence of hundreds or thousands of competing hypotheses. Our results here indicate that this is still the case for less constrained contours. This is an important result, since it shows that a probabilistic framework working exclusively on line segment data, and without any domain-specific knowledge, can be used to obtain a reasonable estimate of the saliency of a compact contour.

5. Experimental Results

Figure 1 shows the result of running our contour extraction procedure on a line-set extracted from the photograph of a car, notice that the line-set is rich in textured components, and shows significant complexity. Within the car itself, there are many possible paths a contour grouping procedure may take to form closed boundaries. The top contours extracted by our algorithm are shown, along with their sub-components. These contours correspond to perceptually significant structure, which illustrates the robustness of the ranking procedure. It is also worth noting that the contours have both smooth sections and sharp bends, which shows that the smoothnes bias does not preclude the formation of groups with sharp corners. In terms of run-time, our current algorithm matches, and in some instances outperforms the more constrained convex-grouping procedure we described in [2], which in turn had been shown to be significantly faster than previous convex grouping schemes. This indicates that the modifications made to the affinity measure, and the introduction of compactness as a grouping principle are indeed capable of managing the combinatorial explosion in the number of possible paths the grouping algorithm has to choose from when forming contours. The reader is encouraged to visit **http://www.cs.utoronto.ca/˜strider/icpr2004/** for additional experimental results.

We have found that it is possible to obtain a further decrease in run-time if we use additional information to bias the search, we are currently experimenting with filters that provide an estimate of the probability that a particular image region is textured, we are using this probability as an additional term in the affinity function, so that search algorithm can avoid grouping through texture-rich regions as long as there is an alternative path. We're also evaluating the use of colour cues during the search phase to increase the robustness of the algorithm and improve the ranking of extracted contours. Preliminary results in these directions look quite promising, and indicate that the algorithm presented here can be easily extended to include other sources of information and grouping constraints.

References

[1] N. A., M. Jordan, and Y. Weiss. On spectral clustering, analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, 2002.

[2] F. J. Estrada and A. D. Jepson. Controlling the search for convex groups. Technical Report CSRG-482, Department of Computer Science, University of Toronto, Jan. 2004.

[3] G. Guy and G. Medioni. Inferring global perceptual contours from local features. *IJCV*, 20(1):113–133, 1996.

[4] G. Guy and G. Medioni. Inference of surfaces, 3d curves, and junctions from sparse, noisy, 3d data. *PAMI*, 19(11):1265–1277, 1997.

[5] D. Huttenlocher and P. C. Wayner. Finding convex groupings in an image. *IJCV*, 8(1):7–29, 1992.

[6] D. W. Jacobs. Robust and efficient detection of convex groups. *PAMI*, 18(1):23–37, 1996.

[7] A. Jepson and R. Mann. Qualitative probabilities for image interpretation. In *ICCV*, pages 1123–1130, 1999.

[8] D. G. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, 1985.

[9] D. G. Lowe. Three dimensional object recognition from single 2-d images. *AI*, 31:355–395, 1987.

[10] S. Mahamud, L. Williams, K. Thornber, and K. Xu. Segmentation of multiple salient closed contours from real images. *PAMI*, 25(4):433–444, 2003.

[11] J. Malik, S. Belongie, T. Leung, and J. Shi. Textons, contours and regions: cue integration in image segmentation. In *CVPR*, pages 918–925, 1999.

[12] R. Mohan and R. Nevatia. Segmentation and description based on perceptual organization. In *CVPR*, pages 333–341, 1989.

[13] R. Mohan and R. Nevatia. Perceptual organization for scene segmentation and description. *PAMI*, 14(6):616–634, 1992.

[14] S. Sarkar and K. L. Boyer. Integration, inference, and management of spatial information using bayesian networks: Perceptual organization. *PAMI*, 15(3):256–274, 1993.

[15] E. Saund. Finding perceptually closed paths in sketches and drawings. *PAMI*, 25(4):475–491, 2003.

[16] J. Shi and J. Malik. Normalized cuts and image segmentation. In *CVPR*, pages 731–737, 1997.

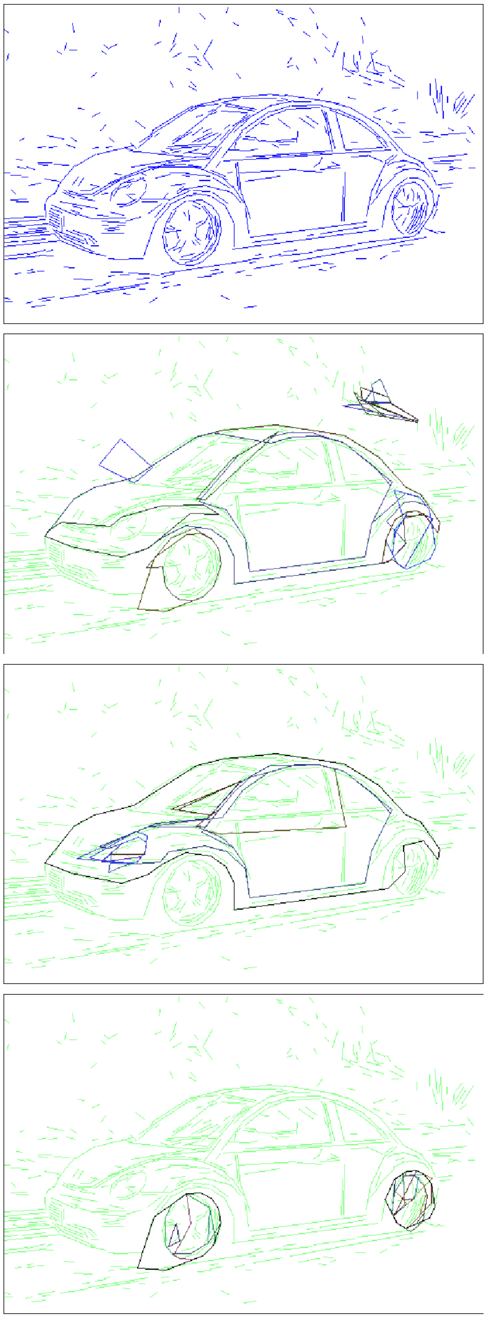

Figure 1. From top to bottom: a) Original lineset (579 non-oriented segments), b) Contours detected at the top level, c) Best contour (black) and its subparts (color), and d) second and third ranked contours (black) and their subparts (color). The algorithm was run with $K = 20$, compactness threshold of $.80$, and normalized affinity threshold of $1.2/K$. Run-time was 21 seconds on a P4, 1.9GHz machine.