Segmentation of Salient Closed Contours from Real Images

Shyjan Mahamud
Dept. of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Karvel K. Thornber
NEC Research Institute, Inc.
4 Independence Way
Princeton, NJ 08540

Lance R. Williams
Dept. of Computer Science
University of New Mexico
Albuquerque, NM 87131

Abstract

Using a saliency measure based on the global property of contour closure, we have developed a method that reliably segments out salient contours bounding unknown objects from real edge images. The measure also incorporates the Gestalt principles of proximity and smooth continuity that previous methods have exploited. Unlike previous measures, we incorporate contour closure by finding the eigen-solution associated with a stochastic process that models the distribution of contours passing through edges in the scene. The segmentation algorithm utilizes the saliency measure to identify multiple closed contours by finding strongly-connected components on an induced graph. The determination of strongly-connected components is a direct consequence of the property of closure. We report for the first time, results on large real images for which segmentation takes an average of about 10 secs per object on a general-purpose workstation. The segmentation is made efficient for such large images by exploiting the inherent symmetry in the task.

1 Introduction

Visual perception evolved in a world of objects many of which are bounded by smooth closed contours. We hypothesize that these contours obey a stochastic distribution which is utilized by perceptual processes in finding contours bounding objects. In prior work [10, 12, 13] this distribution has been modeled and used to derive a saliency measure that exploits the closure of contours bounding objects. It was found that this measure provides a significant improvement over previous approaches in highlighting edges lying on contours bounding objects in small synthetic scenes created from contours of real objects and natural background texture [13]. However, no method was presented for actually segmenting out the salient object contours. Despite the effectiveness of the saliency measure, it will be shown later that a simple threshold on the saliency measure is not sufficient for segmentation, escpecially in cases where two or more object contours have similar saliencies. In this paper, we present a method for segmenting out multiple contours bounding salient objects. Moreover, previously [13] the determination of the saliencies was computationally infeasible for large real images. We have developed an efficient technique that exploits the symmetry inherent in the task, using which, we report for the first time, results on large real images.

Given an edge image as in Fig. 1 (a), we would like to extract out separately the individual contours bounding the two pears. We wish to achieve such a segmentation with no *a-priori* knowledge of the specific objects that generate these contours. Such a task is one of the goals of *perceptual grouping*. In lieu of any specific knowledge about the objects generating the contours, we impose a subset of the Gestalt principles for perceptual organization. Most previous approaches to perceptual grouping of edges have incorporated the local principles of proximity of edges and smooth-continuation of contours in some form or other (e.g., [4, 8]). These methods assume that successive edges are in close proximity and that contours bounding objects are smooth. In addition to these two local properties, we exploit the global property that contours bounding objects must be closed. Unlike proximity and smooth-continuation, closure cannot be reduced to any local property of the contour.

Previous approaches [2, 3] have used graph based search techniques to find closed contours. A graph of affinities between edges is constructed where the affinities model proximity and smooth-continuation. The affinity between two edges is a purely local measure that is proportional to the likelihood that a smooth contour (open or closed) passes through the given edges. Closure is imposed by searching the graph for closed contours while minimizing a global cost function that is related to how salient the closed contours are. Our approach differs from these previous approaches because we first use the local affinity measure (which as noted above does not differentiate between open and closed contours) to compute a global

saliency measure, which is proportional to the relative number of closed contours which join a pair of edges. Such a measure for closed contours was first proposed and compared extensively with previous approaches (including [7, 8, 12]) that do not incorporate closure in [13]. It is only after the computation of this global saliency that we employ a graph search to identify individual closed contours. We will show that in our case, the incorporation of closure in the saliency measure leads naturally to a specific type of graph search, namely, the determination of strongly connected components. This dependency of the specific graph search used on the saliency measure is a distinguishing feature of our work as compared with previous approaches where generic search techniques have been employed that are not dependent on the properties of the specific saliency measures that were used. To illustrate the crucial role played by the global property of closure, we show that a method based on a purely local affinity measure produces poor segmentations.

The determination of the saliency measure requires the solution of an eigen-problem of a matrix that exhibits a special kind of symmetry. Ordinary techniques for the solution of the eigen-problem are infeasible for large real images. We have developed efficient techniques that exploit the special symmetry of the matrix to significantly reduce the time required to compute the eigen-solution. In this paper, we report the first results on real images with a large number of edges. Our technique reduces the time taken to compute the segmentation for each object contour from an average of around 2 1/2 hrs. to around 10 seconds.

2 Problem Formulation

Since the Gestalt principles of proximity and smooth-continuation arise from local properties of the positions and orientations of two edges, we can model them using only local information. Following [10, 12], both of these local properties can be modeled by the distribution of smooth curves that join two given edges. The distribution of curves is modeled by a particle with motion determined by a stochastic process favoring short, smooth, trajectories. Given two directed edges i and j, they determine the probability that a particle starts with the position and direction of edge i and ends with the position and direction of edge j. The particles leave or arrive at an edge at a constant speed γ independent of the edge. The "affinity" from edge i to edge j is denoted by P_{ji} and is the sum of the probabilities of all paths that a particle can take between the two edges (see [10] for details). Essentially, two parameters control the motion of the particle and embody the principles of proximity and smooth-continuation. Each particle has a half-life (τ) which models the principle of proximity. The variance (T) of the directional change of the particle model the principle of smooth-continuation. The speed γ determines the effective scale at which the scene is analyzed, since the affinity between a pair of edges varies with the speed of the particle. At larger speeds, a pair of edges are effectively closer to each other, whereas at slower speeds the same pair are effectively farther apart. In our application we choose a fixed speed that we judge to give a good effective scale for all images.

Because particles need not reach any another edge due to the half-life, in general $\sum_j P_{ji} < 1$. Hence P is not a stochastic (Markov) matrix, and methods based on Markov chains are not applicable. While closed contours do form a Markov chain, the corresponding Markov matrix for the edge saliencies is not known until the edge and link saliencies have been determined.

Smooth-continuity of a curve between two edges implies that the tangent at any point along the curve is continuous. Such tangent-continuity for curves passing between a pair of edges is modeled by the smooth stochastic motion of the particle going from one edge to the other. If we wish to extend the curves to include additional edges then tangent continuity must be enforced at the edges themselves. A contour coming in along a given direction must continue along that direction to satisfy tangent-continuity. This requirement can be ensured by replacing each edge of a given orientation with two oppositely directed edges. A contour can come into and leave a directed edge only along a single direction. If we do not impose tangent-continuity at the edges, we could get contours with cusps (i.e., reversals in direction) at the edges, which are not judged to be salient in practice. For more details see [13]. Since every directed edge i has a sibling edge at the same position but pointing in the opposite direction, it will be convenient to denote the sibling edge by $\bar{i}$.

Imposing tangent-continuity through directed edges has an important implication for the structure of the matrix of affinities P. From symmetry, the probability that any particle travels along a curve starting from edge i and ending in edge j is the same as the probability of a particle traveling the same curve along the reverse direction from edge $\bar{j}$ to edge $\bar{i}$. Hence $P_{ji} = P_{\bar{i}\bar{j}}$. We call this special symmetry of the affinity matrix *reversal-symmetry* which is distinct from the usual symmetry $P_{ji} = P_{ij}$ which need not hold in general. Reversal-symmetry has important implications for both the form of

the expressions which define the saliencies and for efficiently computing them.

In the rest of the paper we will have occasion to associate a vector $\mathbf{s}$ with the set of *directed* edges (e.g. the vector of saliencies for each directed edge), one component for each directed edge. Analogous with the case for edges, a component of such a vector s_i associated with edge i will have a sibling component denoted by $\bar{s}_i = s_{\bar{i}}$ associated with edge $\bar{i}$.

3 Saliency measure

In this section, we first motivate the expression for the saliency measure [13]. We then show that the saliency measure can be determined by the solution to an eigen-problem associated with the affinity matrix P. Given an edge image, we define a closed contour as a finite closed sequence of edges. By a closed sequence we mean that if we start from any edge in the sequence and trace out the contour we will return to the same edge. Each closed contour α has a likelihood (or probability) associated with it, which we denote by $p(\alpha)$ and is the product of the transition probabilities (given by the affinity matrix P) between successive edges in the contour.

3.1 Edge Saliency

We would like to define our saliency measure for an edge to be related to the likelihoods of the various closed contours that thread through that edge. Rather than derive the contribution of each individual closed contour to the saliency for an edge, it is simpler to consider the contribution of the ensemble of all closed contours through that edge by considering the set of infinite closed contours passing through that edge. Each infinite closed contour can be decomposed into a sequence of finite closed contours, and hence the relative likelihood of different infinite closed contours passing through an edge depends on the relative likelihood of the individual finite closed contours that they are composed of. In order to calculate the relative saliencies of infinite contours, we start by considering the relative saliencies of closed contours of finite but large length and take the limit as the length goes to infinity. Restricting ourselves to finite contours for now, the saliency of an edge should be proportional to the expected number of closed contours that pass through that edge. The expected number of closed contours of length n that thread through edge i is simply the sum of the probabilities of all such closed contours :

$$E_i^n = \sum_{\alpha} p(\alpha \mid i \in \alpha, |\alpha| = n) \tag{1}$$

Since we are interested in the relative saliencies of the various infinite contours that thread through different edges, we take the limit $n \to \infty$ for the expected number of closed contours through a given edge i relative to the expected number through all edges and obtain the formal definition for the saliency of edge i :

$$C_i = \lim_{n \to \infty} \frac{E_i^n}{\sum_j E_j^n}$$

With this definition, it turns out that there is a simple relationship between edge saliencies and the eigenvector corresponding to the largest eigenvalue of the affinity matrix P.

Theorem 1 (First Saliency Theorem) *The saliency for edge i is given by :*

$$C_i = s_i \, \bar{s}_i \tag{2}$$

where the s_i's are the components of the eigenvector (normalized such that $\sum_i s_i \bar{s}_i = 1$) corresponding to the largest eigenvalue λ in magnitude of the affinity matrix $\mathbf{P}$, i.e. $\mathbf{Ps} = \lambda \mathbf{s}$.

Proof. See [5] and also [13] for an earlier proof.

Note that due to reversal-symmetry, we would expect $C_i = C_{\bar{i}}$ as can be verified from the expression above.

3.2 Link Saliency

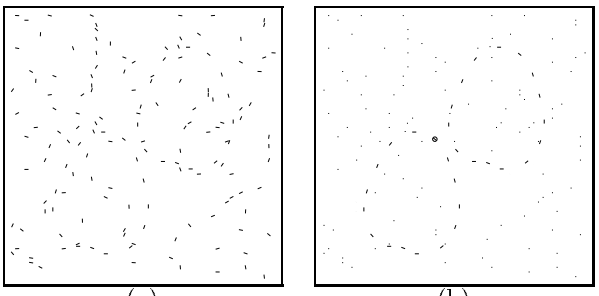
For the purpose of segmentation, in addition to the edge saliencies, we will also need information that will help us trace out contours given a starting edge. Specifically, given two edges j and i we would like to know the relative likelihood that closed contours pass through edges j and i successively. We define the *link saliency* C_{ij} to be the relative saliencies of the closed contours that pass through edges j and i successively. Analogous to the definition for the edge saliencies, we have :

$$C_{ij} = \lim_{n \to \infty} \frac{E_{ij}^n}{\sum_l E_l^n}$$

where E_{ij}^n is the expected number of closed contours of length n that go through edges j and i successively, and E_l^n is as defined before in (1). The link saliencies also turn out to have a simple relationship with the eigenvector corresponding to the largest eigenvalue of P.

Theorem 2 (Second Saliency Theorem) *The link-saliencies between any two edges j and i are given by :*

$$C_{ij} = \frac{\bar{s}_i P_{ij} s_j}{\lambda} \tag{3}$$



(a) (b)

Figure 1: (a) An example edge image created from two copies of a real pear contour superimposed on a background texture (b) Saliency plot.

where the s_i's are the components of the eigenvector (normalized such that $\sum_i s_i \bar{s}_i = 1$) corresponding to the largest eigenvalue λ of the affinity matrix P.

Proof. See [5]. It should be noted that the notion of the edge saliencies C_i's was first introduced in [13]. However, the notion of link saliencies and the above theorem is new and has not been presented before.

Again, as in the case of the edge saliencies, due to reversal-symmetry we would expect $C_{ij} = C_{\overline{j}\,\overline{i}}$ as can be verified from the expression above (recall, $P_{ij} = P_{\overline{j}\,\overline{i}}$ and $\bar{\bar{s}}_i = s_i$).

Since we are concerned with closed contours, an important conservation property holds for all edges. Any *closed* contour that goes from some edge k into a second edge i must continue onto some third edge j. This is not necessarily true in the case of open contours. We confirm this conservation property and at the same time use it as a consistency check on the expressions for the C_{ij}'s and C_i's :

$$\sum_k C_{ik} = \sum_k \frac{\bar{s}_i (P_{ik} s_k)}{\lambda} = \frac{\bar{s}_i (\lambda s_i)}{\lambda} = \bar{s}_i s_i = C_i$$

Doing a similar calculation for $\sum_j C_{ji}$, we find

$$\sum_k C_{ik} = C_i = \sum_j C_{ji}$$

We conclude this section by demonstrating how well our saliency measure performs for the two-pear example of Fig. 1. The saliency measure for each edge in the figure was determined through the expressions for C_i in equation (2) after solving for the largest eigenvalue of P and its corresponding eigenvector. The saliency plot is shown in Fig. 1 (b). The length of an edge in the plot is proportional to its saliency. As can be seen, the edges bounding both pears have high (and comparable) saliencies. The saliencies of all other edges have been suppressed (numerically, their saliencies are 20 orders of magnitude smaller than those of the pears).

In order to separately identify the two contours bounding the pears, we might think of simply thresholding the saliencies. However, as illustrated in this example (and in general), it is possible for such a simple thresholding scheme to group together edges bounding distinct objects. In the next section, we develop a more robust approach that uses the link saliencies C_{ij}'s to group together the set of edges that belong to distinct objects.

4 Segmentation

The goal of segmentation is to group together in distinct sets, edges bounding distinct objects in the scene. To motivate our segmentation algorithm, consider the hypothetical case where some oracle provided us with a set S of closed contours in the scene whose saliencies are above some threshold. We can construct a graph whose vertices correspond to the edges in our scene. We create a directed link in this graph from edge i to edge j if i and j are successive edges in some salient contour in S. It is proved in [5] that such a construction induces a partition of the graph into a set of isolated *strongly-connected* components. A strongly-connected component [1] is a set of edges in which any pair of edges i and j have a path from one to the other, i.e., $i \rightsquigarrow j$ as well as $j \rightsquigarrow i$. In general each strongly-connected component will contain multiple salient contours that share common edges. It is shown in [5] that the partition into a set of strongly-connected components is a direct consequence of the property of closure of the contours in S. As noted in the introduction, the strong dependence between the nature of the partition and the property of closure is a distinguishing feature of our approach, as compared with other approaches [2, 3] that employ generic graph search techniques that are not dependent on any specific property of the saliency measures used. More precisely, in our approach, the determination of strongly-connected components makes sense only in the context of using a saliency measure that incorporates closure.

In practice, of course, we do not know the salient contours beforehand. Nevertheless, since the links in the salient contours become the links in the graph, all we need to know is which of the links are salient, i.e. the likelihood that some salient contour passes through a given link. The *link-saliencies* (C_{ij}'s) provide precisely an encoding of such information.

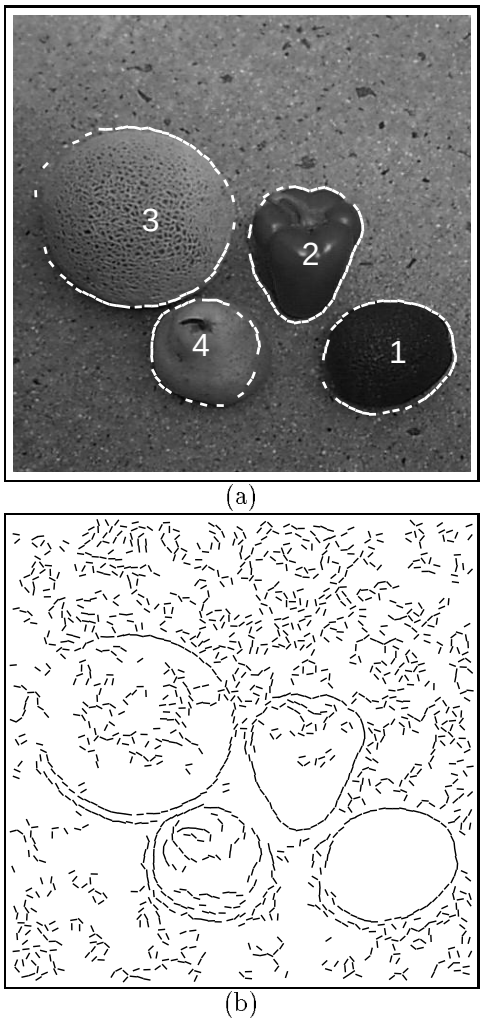
Ideally, the set of edges will be partitioned into

isolated components. However, in practice not all of the components provide reliable segmentations. The dominant contours tend to suppress the saliencies of all other contours to such an extent that the saliencies of these non-dominant contours are not sufficient to induce components that can be isolated reliably. Hence in practice, we begin by extracting the most salient contours. Since such contours will normally pass through the most salient edge, we first identify the contours corresponding to the strongly-connected component containing the most salient edge. Having identified the most salient contours, we suppress their link saliencies in order to reveal the next set of dominant contours. We suppress the current set of dominant contours by deflating the affinities of *all* links among the edges in the strongly-connected component. Specifically, if i and j are edges in the component, then the link $i \rightarrow j$ is deflated by setting $P_{ji} = 0$ (as well as setting the reversal-symmetric "sibling" $P_{\bar{i}\bar{j}} = 0$). We then iterate this process to reveal multiple salient contours.

Again considering the ideal case, the strongly-connected component containing the most salient edge will be isolated from the other components. In practice, due to noise, some of the C_{ij}'s might wrongly indicate that the strongly-connected component containing the most salient edge is connected to one or more of the other strongly-connected components. Nevertheless, we can extract the component of interest by utilizing an important property of strongly-connected components–the set of edges in a strongly-connected component containing a given edge is the intersection of the set of edges reachable from the given edge and the set of edges reachable from the same edge when all the links have their directions reversed [1]. In our case, due to reversal-symmetry, the above property reduces to a particularly simple form. In order to identify the strongly-connected component containing the most salient edge i, we find the intersection of the set of undirected edges reachable from edge i and the set of undirected edges reachable from $\bar{i}$ (note that both i and $\bar{i}$ have the same saliency as discussed in section 3).

5 Results

We show results of our segmentation algorithm on a few real images. All the images were taken using a Kodak DC50 480x480 pixel digital color camera. The Canny edge detector was run on the images after converting them to greyscale, with the parameters $\sigma = 3.0$, low hysteresis threshold = 0.2 and high hysteresis threshold = 0.8. The set of edges returned by the Canny edge detector were found to be quite redundant. The edges are sampled to improve running times with almost no sacrifice in performance. In our experiments we sample the edges such that no two edges are closer than 5 pixels apart.



(a)

(b)

Figure 2: Fruits on concrete. (a) Greyscale image with the segmented contours numbered in the order they are extracted (b) Canny edge output.

The entries of the affinity matrix $\mathbf{P}$ were calculated with parameter settings (see § 2 for their descriptions and also [10]) $\gamma = 0.15$, $T = 0.004$ and $\tau = 5.0$. All edge images are remapped to a 64×64 image size. Since the affinity matrix $\mathbf{P}$ has a special symmetry (the reversal symmetry), we had previously developed an algorithm that finds the eigenvector corresponding to the largest eigenvalue of $\mathbf{P}$ (required for the computation of our saliency measures) by exploiting the reversal symmetry. See [11] for details. In our first example we chose a simple scene where non-occluding objects (fruits) were placed on a textured background

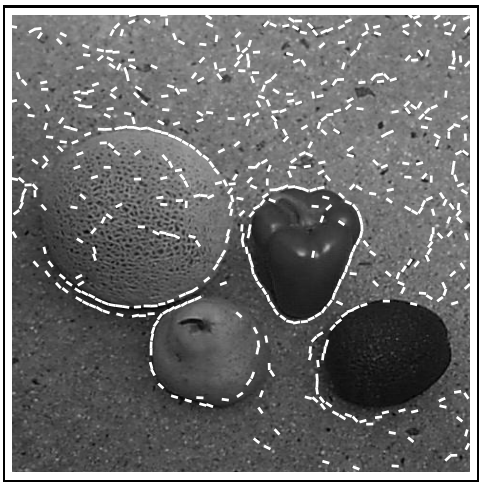


Figure 3: Fruits on concrete. When the C_{ij}'s have been replaced by P_{ij}'s the segmentation during the second iteration breaks down (see text)

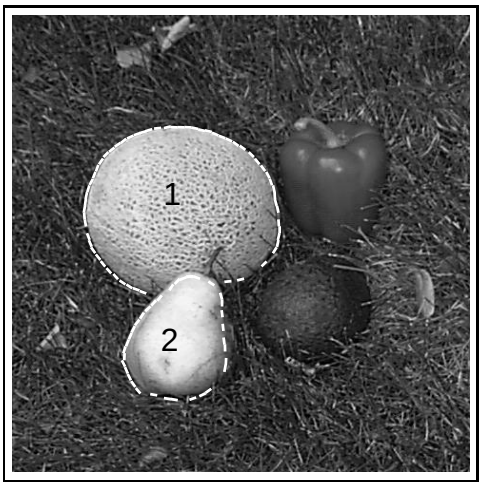


Figure 4: Fruits on grass. Greyscale image with segmentations superimposed.

(concrete). Fig. 2 shows four fruits on a concrete background in greyscale (a) and the corresponding edge image (b) (with 2800 directed edges after the sampling process described above). Notice that the contrast between the texture of the fruit on the top-left (a cantelope) and that of the background is quite low. As a result, few edges are detected along some parts of the boundary of the cantelope. Superimposed on Fig. 2 (a) are the contours which are identified during the successive iterations of the segmentation algorithm. It is interesting to note that the contour bounding the cantelope has been extracted despite the fact that there are large gaps in some parts of the contour. We next show the importance of the global information encoded by our link saliencies C_{ij}'s for segmentation by replacing them with the P_{ij}'s that encode only local information. The edge saliencies C_i's are left unchanged. With this replacement, the segmentation algorithm extracts out the same contour in the first iteration as the original algorithm with the C_{ij}'s. Note that this contour is easy to trace out since there are no large gaps present between successive edges of the contour. However, the hard part is to get a starting edge (i.e., the most salient edge in the current iteration) which (for this demonstration) is still being provided by the C_i's. Fig. 3 shows the segmentation after the second iteration. As can be seen, the segmentation completely breaks down. The P_{ij}'s are sufficient as long as we start off from the most salient edge in each iteration and there are no large gaps in the contours being traced. The breakdown in the second iteration shows the need for the more global information encoded in the C_{ij}'s in cases where there are large gaps in the contours being traced.

The eigen-solver for the matrix $\mathbf{P}$ described above (see [11] for details) is adaptive, the time roughly varying according to the complexity of the contours extracted and the number of edges in it. As expected the first iteration took the least time of 3 sec since the contour extracted is relatively simple. The third iteration took the longest time of 20 sec possibly because of the large gaps in the contour being extracted (bounding the cantelope). The average time for the 4 iterations is 9.9 sec. Fig. 4 shows the same four fruits with grass as the background and with one of the fruits occluding another. Due to poor contrast between the two dark fruits and the background the Canny edge detector does not reliably detect the edges bounding the two fruits. The fruits are hardly salient in the edge image (not shown) even for human observers. Our algorithm can be expected to extract out contours only when provided with reliable edge information. In this case the algorithm picks out only the other two fruits in the image. Of the two fruits that it does pick out, one partly occludes the other. Due to the poor contrast between the two fruits, the edge information (especially the orientation) is quite poor in the region around the occlusion. However, despite this fact, and the fact that the contour bounding the occluded fruit contains a large gap at the occlusion, the algorithm segments out both fruits individually. Finally, Fig. 5 shows an example where there are significant shadows (around the stones) which produce strong smooth contours. However, since they are open contours, they are not as salient as the closed contours bounding the stones and hence do not confuse the algorithm.

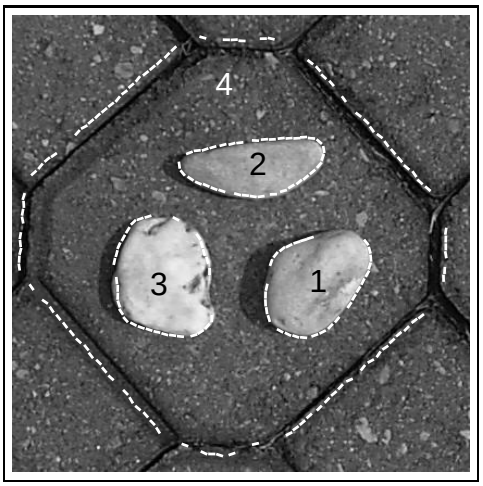


Figure 5: Stones on pavement. Greyscale image with segmentations superimposed.

6 Conclusion

We have demonstrated the usefulness of a saliency measure based on the global property of contour closure in segmenting out multiple closed contours from real images. We have shown the importance of the global information encoded by our link saliencies C_{ij}'s for segmentation as opposed to using just the P_{ij}'s that encode only local information.

Our approach to grouping edges into salient contours involves the solution of an eigen-problem. Recently, other approaches [6, 9, 7] have also proposed grouping image features by solving a corresponding eigen-problem. The normalized minimum-cut approach proposed in [9] can group more general image features than our approach can. However, since we restrict ourselves to grouping edges into salient closed contours, we are able to impose the important constraint of edge-directionality that is not applicable for other features like pixel intensities. Also, imposing edge-directionality results in a non-symmetric affinity matrix $\mathbf{P}$ for which the min-cut approach proposed in [9] does not apply. As noted in § 2, we can get a symmetric affinity matrix if we do not impose edge-directionality, in which case the min-cut approach would apply. However, the symmetric affinity matrix permits contours with cusps at the edges to be salient (see the discussion in § 2). Hence, we would expect poor performance with a min-cut approach when edge-directionality is not imposed. The dominant eigenvector approach that was proposed in [6, 7], while applicable to more general features than edges, also applies only to symmetric matrices and hence cannot exploit edge-directionality. Summarizing, since in this paper, we are interested in finding out how much can be accomplished using only edge information, methods that exploit constraints from the specific domain, such as ours, are expected to give better results compared with more general-purpose methods.

The segmentation algorithm assumes a fixed value for the speed γ of the particle modeling the distribution of curves between two given edges. Equivalently, this fixes the effective scale or size for the input image (see § 2). A more principled approach would be able to extract out a contour regardless of the rate at which its edges are sampled. A straightforward approach would be to sweep the scale (by varying the speed γ) in each iteration and find the most salient contour across all scales (see [11]). Contours with different samplings would become dominant at different scales. We are currently investigating this approach and expect it to result in improved, scale-invariant segmentations.

References

[1] Cormen, T.H., Leiserson, C.E. and Rivest, R.L. 1989. *Introduction to Algorithms*, Chapter 23, MIT Press.

[2] Elder, J.H. and Zucker, S.W. 1996. Computing contour closure. In *ECCV '96*, Cambridge, UK. **Vol. I**:14–18.

[3] Jacobs, D. 1993. Robust and Efficient Detection of Convex Groups. In *CVPR*, 770–771.

[4] Lowe, D.G. 1985. *Perceptual Organization and Visual Recognition*, Kluwer, Boston.

[5] Mahamud, S., Thornber, K.K. and Williams, L.R. 1998. Extracting Multiple Salient Closed Contours from Real Images. *NEC Tech. Report 98-120*

[6] Perona, P. and Freeman, W. 1998. A Factorization Approach to Grouping. In *ECCV*, Freiburg, Germany.

[7] Sarkar, S. and Boyer, K. 1996. Quantitative Measures for Change based on Feature Organization: Eigenvalues and Eigenvectors, In *CVPR '96*, San Fransisco, CA.

[8] Shashua, A. and Ullman, S. 1988. "Structural Saliency : The Detection of Globally Salient Structures Using a Locally Connected Network", In *ICCV*, FL.

[9] Shi, J. and Malik, J. 1997. Normalized Cuts and Image Segmentation. In *CVPR '97*, Puerto Rico, USA.

[10] Thornber, K.K. and Williams, L.R. 1996. Analytic Solution of Stochastic Completion Fields. *Biol. Cybern.* **75**:141–151.

[11] Thornber, K.K., Mahamud, S. and Williams, L.R. 1998. The Eigenvalue Problem for Reversal Matrices. *NEC Tech. Report 97-162*

[12] Williams, L.R. and Jacobs, D.W. 1997. Local Parallel Computation of Stochastic Completion Fields. *Neural Computation* **9**:859–881.

[13] Williams, L.R. and Thornber, K.K. 1998. A Comparison of Measures for Detecting Natural Shapes in Cluttered Backgrounds. In *ECCV*, Freiburg, Germany.