Segmentation of Multiple Salient Closed Contours from Real Images

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

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

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

Kanglin Xu
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 incorporates the Gestalt principles of proximity and smooth continuity that previous methods have exploited. Unlike previous methods, we incorporate contour closure by finding the eigenvector with largest positive real eigenvalue of a matrix defining a stochastic process which 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.*

I. 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 [22], [24], [25] this distribution has been modeled and used to derive a saliency measure (termed WT) which 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[25]. However, no method was presented for actually segmenting out the salient object contours. Despite the effectiveness of the WT measure, we will show that a simple threshold on the raw saliency values is not sufficient for segmentation, especially in cases where two or more object contours have similar saliencies. In this paper, we present a method for extracting multiple salient closed contours bounding distinct objects. Previously, a routine from a standard numerical library was used to solve the eigenproblem required to compute the WT measure. However, due to the number of edges involved, this is infeasible for large real images. Consequently, in this paper, we have developed an efficient technique that exploits the sparseness and symmetry of representations intrinsic to the problem, using which, we report for the first time, results on large real images.

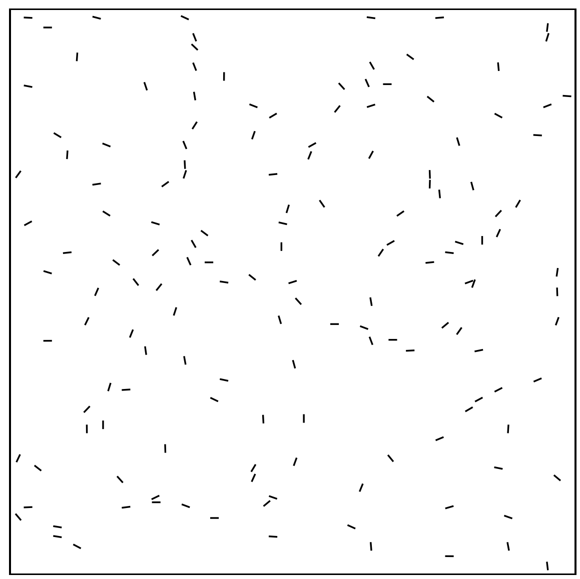


Fig. 1. An example edge image. The image was synthetically created by superimposing two copies of edges from the boundary of a real pear on a background texture.

Given an edge image as in Fig. I, 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 Gestalt principles of proximity and good continuation in some form (*e.g.*, [2], [9], [6], [12]). These methods assume that adjacent edges of an object boundary are close together and can be smoothly interpolated. In addition to these two local properties, we exploit the global property that contours bounding objects must be closed. Unlike proximity and good continuation, closure cannot be reduced to a local property defined for pairs of edges in isolation.

Previous approaches[1], [6], [11], [16] have used graph based search techniques to find closed contours. A graph of affinities between edges is constructed where the affinities model proximity and good continuation. The affinity between two edges is a purely local measure which is proportional to the likelihood that a smooth contour (open or closed) passes through a pair of edges. Closure is imposed by searching the graph for closed contours while minimizing a global cost function which 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 [9], [17], [20]) which do not incorporate closure in [25]. Only after the computation of this global saliency measure do we employ a graph search technique to identify individual closed contours. We show that using a saliency measure based on contour closure leads naturally to a specific type of graph search, namely, the determination of *strongly connected components*. The close relationship between the strongly connected component computation and the closure property of the global saliency measure distinguishes our work from previous approaches, where generic graph search techniques have been applied to graphs representing a local affinity measure defined for pairs of edges in isolation. To illustrate the crucial role played by the global property of contour closure, we show that a method based on a purely local affinity measure produces poor segmentations.

Computing the saliency measure requires identifying the eigenvector with largest positive real eigenvalue of a sparse, positive matrix exhibiting a specific kind of symmetry. Ordinary techniques for the computation of eigenvectors and eigenvalues are infeasible for large real images. We have developed efficient techniques which exploit the sparseness and symmetry of the matrix to significantly reduce the time required to compute this eigenvector. 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.

II. Problem Formulation

Since the Gestalt principles of proximity and good continuation can be reduced to local properties of the positions and orientations of two edges, we can model them using only local information. Following[14], [22], [24], proximity and good continuation can be modeled by a distribution of smooth curves traced by particles moving with constant speed in directions undergoing Brownian motion. In our work, the "affinity" between edge i and 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 [22] for details). Two parameters control the motion of the particle and embody the Gestalt principles of proximity and good continuation. Each particle has a half-life, τ, which determines the distance over which pairs of edges are likely to be linked by random walks. Hence, τ models proximity. The variance, T, of the Gaussian random variable representing change in direction models the principle of good continuation. A third parameter, γ, represents the speed of the particle, and hence determines the effective scale at which the scene is analyzed, since the affinity between pairs of edges varies with speed. At larger speeds, the distance between a pair of edges is effectively smaller, while at slower speeds the same distance is effectively larger. In our initial experiments, we chose a fixed speed that was judged to give good results for most images. In a later section, we present results where the optimal γ for each object in the scene is identified using an optimization method. This leads to a scale-invariant segmentation.

Because particles at one edge need not reach another edge due to the half-life, in general $\sum_j P_{ji} < 1$. Hence $\mathbf{P}$ is not a stochastic (Markov) matrix, and methods based on Markov chains are not directly applicable. While closed contours do form a Markov chain (see [26]), the Markov matrix $\mathbf{M}$ is not known until the edge and link saliencies have been determined, and these are functions of the eigenvector with largest positive real eigenvalue of $\mathbf{P}$.

The smooth continuation of a curve between two edges requires that the tangent at any point along the curve be continuous. If we wish to extend the curves to include additional edges, then tangent continuity must be enforced at the edges themselves. A particle visiting an edge, and traveling in a given direction, must continue along in that same direction to preserve tangent continuity. This requirement can be ensured by replacing each *oriented* edge, where the orientation is an angle in the range, $[0-\pi)$, with two oppositely directed edges, where the directions are angles in the range, $[0-2\pi)$. A particle must enter and exit a directed edge in the same direction. If we do not impose tangent continuity at the edges, it is possible to 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 [25]. 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 $\bar{\imath}$.

Imposing tangent continuity through the use of directed edges has an important implication for the structure of the matrix of affinities $\mathbf{P}$. 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 from edge $\bar{\jmath}$ to edge $\bar{\imath}$ in the reverse direction. Hence $P_{ji} = P_{\bar{\imath}\bar{\jmath}}$. 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 the problem of 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 $\bar{s}_i = s_{\bar{\imath}}$ associated with edge $\bar{\imath}$.

III. Saliency measure

In this section, we first motivate the expression for the WT saliency measure introduced in [25]. We then show that the WT saliency measure can be computed by solving an eigenproblem associated with the affinity matrix $\mathbf{P}$. 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 $p(\alpha)$. This probability is the product of the transition probabilities (given by the affinity matrix $\mathbf{P}$) between successive edges of the contour.

A. Edge Saliency

We would like to define the saliency of an edge such that it is directly related to the likelihood that a closed contour contains that edge. Rather than derive the contribution of individual closed contours to the saliency of an edge, it is simpler to consider the contribution of the ensemble of all closed contours through that edge. We begin by considering the set of infinite closed contours containing the 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 containing an edge depends on the relative likelihood of the individual finite closed contours of which they are composed. In order to calculate the relative saliencies of infinite closed contours, we start by considering the relative saliencies of closed contours of finite 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 which contain that edge. The expected number of closed contours of length n which contain edge i is simply the sum of the probabilities of all such closed contours :

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

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

$$c_i = \lim_{n\to\infty} \frac{E_i^n}{\sum_j E_j^n} \quad (2)$$

This definition suggests that there is a simple relationship between edge saliencies and the eigenvector corresponding to the largest positive real eigenvalue of the affinity matrix, **P**.

Theorem 1—First Saliency Theorem: The saliency for edge i is given by :

$$c_i = s_i \, \bar{s}_i \quad (3)$$

where the s_i's are the components of the eigenvector (normalized so that $\sum_i s_i \bar{s}_i = 1$) corresponding to the largest positive real eigenvalue, λ, of the affinity matrix **P**, *i.e.* $\mathbf{Ps} = \lambda \mathbf{s}$.
Proof. See Appendix A and also [25] for an earlier proof.

It is important to note that since **P** is *positive* (all entries are positive), Perron's theorem[10] guarantees that the largest eigenvalue of **P** will be real and positive. The components of the corresponding eigenvector s_i will all be positive (*i.e.*, $s_i > 0$). Note that due to reversal-symmetry, we would expect $c_i = c_{\bar{i}}$ as can be verified from the expression above.

B. Link Saliency

For the purpose of segmentation, in addition to the edge saliencies, we will also need information which will help us trace out contours given a starting edge. Specifically, given two edges, j and i, we would like to know the probability that a closed contour passes through edge j and then, without visiting another edge, through edge i. We define the *link saliency*, C_{ij}, to be the relative number of closed contours that pass through edges j and i in succession. Analogous to the definition for edge saliency, we have :

$$C_{ij} = \lim_{n\to\infty} \frac{E_{ij}^n}{\sum_l E_l^n} \quad (4)$$

where E_{ij}^n is the expected number of closed contours of length n which pass through edges j and i in succession, and E_l^n is as defined before in (1). Like the edge saliencies, the link saliencies also have a simple relationship with the eigenvector corresponding to the largest positive real eigenvalue of **P**.

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

$$C_{ij} = \frac{\bar{s}_i P_{ij} s_j}{\lambda} \quad (5)$$

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 positive real eigenvalue, λ, of the affinity matrix, **P**.
Proof. See Appendix A.

As in the case of the edge saliencies, due to reversal symmetry, we would expect $C_{ij} = C_{\bar{j}\bar{i}}$ as can be verified from the expression above (recall that $P_{ij} = P_{\bar{j}\bar{i}}$ 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 to a second edge i must continue on to 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 :

$$\begin{aligned} \sum_k C_{ik} &= \sum_k \frac{\bar{s}_i (P_{ik} s_k)}{\lambda} && (6) \\ &= \frac{\bar{s}_i (\lambda s_i)}{\lambda} && (7) \\ &= \bar{s}_i s_i && (8) \\ &= c_i. && (9) \end{aligned}$$

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

$$\sum_k C_{ik} = c_i = \sum_j C_{ji}. \quad (10)$$

C. Contour Saliency

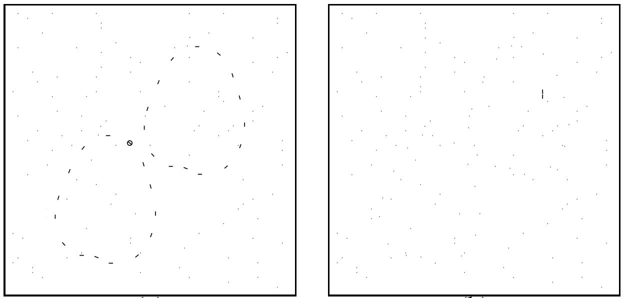
Ideally, our segmentation algorithm should extract closed contours in order of increasing saliency. A possible definition for the saliency of a closed contour would be to define it as the probability of a particle tracing a path through the same edges, *i.e.* the product of the the affinities along the contour's path. However, this definition is dependent on the length of the contour. A closed contour, α, and another closed contour formed by traversing the edges in α twice, *i.e.*, $\alpha \cdot \alpha$, should be judged to have the same saliency. However, it is clear that the probability of the second contour will be much less than that of the first. In fact, it will be the square of the first. A more natural definition for the saliency of a closed contour, α, a definition which is invariant to repetition, is the geometric mean of the affinities along the contour's path:

$$\lambda(\alpha) = p(\alpha)^{1/|\alpha|} \quad (11)$$

where $|\alpha|$ is the length of the closed contour and $p(\alpha)$ is the product of the affinities which comprise it. In other words, if the length normalized probability of one contour is greater than that of a second contour, then we consider the first contour to be more salient than the second. This definition of contour saliency has an interesting relationship with the affinity matrix, **P**, constructed from the given contour (see [25]). If we imagine a scene containing just the closed contour, α, and where the probabilities between non-adjacent edges are zero, then the saliency of the contour is just the largest positive real eigenvalue of **P**.

D. Importance of Directionality

We conclude this section by demonstrating how well the WT saliency measure performs for a simple example consisting of edges from the silhouettes of two pears artificially superimposed on a background texture. See Fig. I. The saliency measure for each edge in was computed using the expressions for c_i inequation (3) after solving for the largest positive real eigenvalue of **P** and its corresponding eigenvector. The saliency plot is shown in Fig. 2 (a). The length of an edge in the plot is proportional to its saliency. It can be plainly seen that 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.



(a) (b)

Fig. 2. Saliency plots for the 2-pear example. (a) Our measure with directed edges (b) Our measure with undirected edges. The length of each edge is proportional to its saliency value.

Using the same example, we demonstrate the importance of using pairs of directed edges to form an affinity matrix, **P**, of size $2N \times 2N$ as opposed to simply using the N edges to form a symmetric affinity matrix, **A**, of size $N \times N$. Recall that this mechanism is required so that closed contours do not include reversals in direction at the locations of the edges. For the purpose of this demonstration, we construct a symmetric affinity matrix **A** from **P** by setting $A_{ij} = P_{ij} + P_{i\bar{j}} + P_{\bar{i}j} + P_{\bar{i}\bar{j}}$. It can be verified that **A** is symmetric because $P_{ij} = P_{\bar{j}\bar{i}}$. Fig. 2 (b) shows the squared magnitude of the components of the eigenvector with largest positive real eigenvalue of **A**. Two edges in the background texture which, simply by chance, are proximal and very nearly collinear, are extremely salient while edges forming the closed boundary of the pears are ignored. It follows that using a non-symmetric affinity matrix, **P**, and pairs of oppositely directed edges, i and $\bar{\imath}$, is essential to satisfactory performance of the saliency measure.

In order to distinguish the contours bounding the two pears, one might try to simply threshold the edge saliencies, *i.e.*, the c_i's. However, as is illustrated by this example, edges from different objects can have saliencies of comparable magnitude. It is therefore likely that such a simple strategy will group together edges bounding distinct objects. In the next section, we develop a more robust approach that uses the link saliencies, *i.e.,* the C_{ij}'s, to group together sets of edges belonging to individual objects.

IV. Segmentation

The goal of segmentation is to group together into distinct sets, edges bounding distinct objects in the scene. To motivate our segmentation algorithm, consider the hypothetical case where some oracle provides 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 of some salient contour in S. The Third Saliency Theorem (see Appendix A) tells us that such a construction induces a partition of the graph into a set of isolated *strongly-connected* components. A strongly-connected component[5] 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 the Appendix A 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 [6], [11] which employ generic graph search. More precisely, in our approach, the determination of strongly-connected components makes sense only in the context of a graph derived using a saliency measure based on contour 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, *i.e.*, the C_{ij}'s, encode precisely this 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 the degree that the saliencies of these non-dominant contours are insufficient to induce components that can be isolated reliably. Hence, in practice, we begin by extracting the most salient contours, and since such contours will normally contain 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 $i \to j$ is deflated by setting $P_{ji} = 0$ (as well as setting the reversal-symmetric "sibling" $P_{\bar{\imath}\bar{\jmath}} = 0$). We then iterate this process to reveal multiple salient contours.

Ideally, 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 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 if all links are reversed.* Because of reversal symmetry, the above property reduces to a particularly simple form. Let $reachable(j)$ be the set of edges reachable from a given edge j. Due to reversal symmetry it can be verified that the set of edges reachable from j when all links are reversed is the same as the *reversal* of the set of edges reachable from edge $\bar{\jmath}$. The reversal of the set $reachable(\bar{\jmath})$ is defined to be

$$\overline{reachable}(\bar{\jmath}) = \{\bar{k} \mid k \in reachable(\bar{\jmath})\}.$$

Hence, in order to identify the strongly-connected component containing the most salient edge j, we find

$$reachable(j) \cap \overline{reachable}(\bar{\jmath}).$$

See Algorithm 1. Interestingly, the above is analogous to the expression for edge saliency, $c_i = s_i \bar{s}_i$. One needs simply to replace the the eigenvector s_i with the set $reachable(i)$, the reversal operator for vectors with the reversal operator for sets, and component-wise multiplication of vectors with intersection of sets. In our case, due to reversal symmetry, the above property reduces to a particularly simple form.

Algorithm 1: Extract an object.

$extract(A)$
begin
  $\mathbf{P} \leftarrow affinities(A)$
  $\mathbf{s} \leftarrow eigenvector(\mathbf{P})$
  $j = \operatorname*{argmax}_i \; (\bar{s}_i \; s_i)$
  return $reachable(j) \cap \overline{reachable}(\bar{\jmath})$
end

In order to decide if a link is salient or not, we need to threshold the C_{ij}'s. We could use a single threshold for the entire graph. However, we can do better by choosing an adaptive threshold for the set of links which originate from edge j, *i.e.*, the j-th column of $\mathbf{C}$. To threshold these C_{ij}'s in a natural manner, we sort the j-th column in decreasing order. In this sorted list, $\mathbf{l}$, we find the k-th largest value (in all of the experiments in this paper k equals two). Edges joined by links from j with magnitude larger than the l_k are assumed to lie on salient closed contours, and are therefore selected. Such a thresholding scheme might misclassify certain links as salient. However, we have observed that the extraction of strongly-connected components is usually robust to such misclassifications. See Algorithm 3.

The termination criterion we use for the current implementation is to simply stop after reporting some fixed number of components. See Algorithm 2. One way to justify the use of such a simple criterion is to imagine a higher-level module that, for example, performs object-recognition, and which employs the segmentation algorithm to highlight regions where the presence or absence of some object can be determined using domain knowledge available to the recognizer. It is up to the recognizer to determine the number of salient contours that it wants to process (based on, for example, real-time constraints). If the module determines that the segmentation algorithm is reporting garbage after a certain number of iterations, then it can decide to terminate the search for additional contours. Alternatively, we could stop when the largest positive real eigenvalue, λ, becomes negligibly small.

Algorithm 2: Segment an image.

$segment(A)$
begin
  $S \leftarrow \emptyset$
  for $i \leftarrow 1$ **to** N **do**
    $B \leftarrow extract(A)$
    $S \leftarrow S \cup \{B\}$
    $A \leftarrow A - B$
  end
  return S
end

Algorithm 3: Compute edges reachable from j.

$reachable(j)$
begin
  $B \leftarrow \{j\}$
  if not $visited(j)$ **then**
    for $i \leftarrow 1$ **to** $|A|$ **do**
      $C_{ij} \leftarrow \bar{s}_i \; P_{ij} \; s_j$
    end
    $\mathbf{l} \leftarrow sort(column(\mathbf{C}, j))$
    for $i \leftarrow 1$ **to** $|A|$ **do**
      if $C_{ij} > l_k$ **then**
        $B \leftarrow B \cup reachable(i)$
      end
    end
    return B
  end
end

As a demonstration, we apply the segmentation algorithm to the two pear example from Fig. I. The segmentation in the first and second iterations are shown in Fig. 3 (a) and (b) respectively. As previously noted, a segmentation based on simply thresholding the edge saliencies would not be able to separate the two pears.

V. Results

In this section, we show results of our segmentation algorithm on a few real images. All the images were taken using a

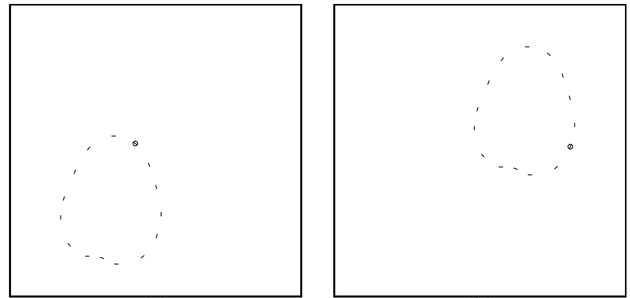

(a) (b)

Fig. 3. Segmentation results for the two pear example. (a) First iteration. (b) Second iteration.

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.

The entries of the affinity matrix $\mathbf{P}$ were calculated with parameter settings (see Section II for their descriptions and also [22]) $\gamma = 0.15$, $T = 0.004$ 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 positive real eigenvalue of $\mathbf{P}$ (required for the computation of the WT saliency measure) by exploiting the reversal symmetry. See [23] for details.

A. Example Segmentation

In our first example we chose a simple scene where non-occluding objects (fruits) were placed on a textured background (concrete). Fig. 4 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. Fig 4 (a),(c),(e), (g), and (i) show the edge saliencies, *i.e.* the c_i's, computed during the first five iterations of the segmentation algorithm. Fig 4 (b),(d),(f),(h), and (j) show the corresponding contours which are extracted during those same iterations. 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.

Fig. 4 (a) shows the variation of the saliency of the dominant contour across iterations. The dominant contour is extracted at each iteration by tracing out the most salient links starting from the most salient edge until we return to the most salient edge again. Its saliency is measured by the expression in equation (11). Since the saliency of the dominant contour decreases as we extract out successive contours, we see that the contours are indeed extracted in the order of their saliencies.

Finally, we give the time requirements for our algorithm for this example. It takes $\approx$ 13 seconds to generate a total of $\approx 111,000$ entries in the sparse affinity matrix $\mathbf{P}$ on an SGI R10000. Since generating the entries in the matrix is easily parallelizable, it is useful to know the time per entry which is $\approx$ 0.11 msec. In Fig. 4 (b), we show the time taken to isolate successive objects. The eigensolver described above (see [23] for details) is adaptive, the time roughly varying according to the complexity of the contours extracted and the number of edges each contains. As expected, the first iteration took the

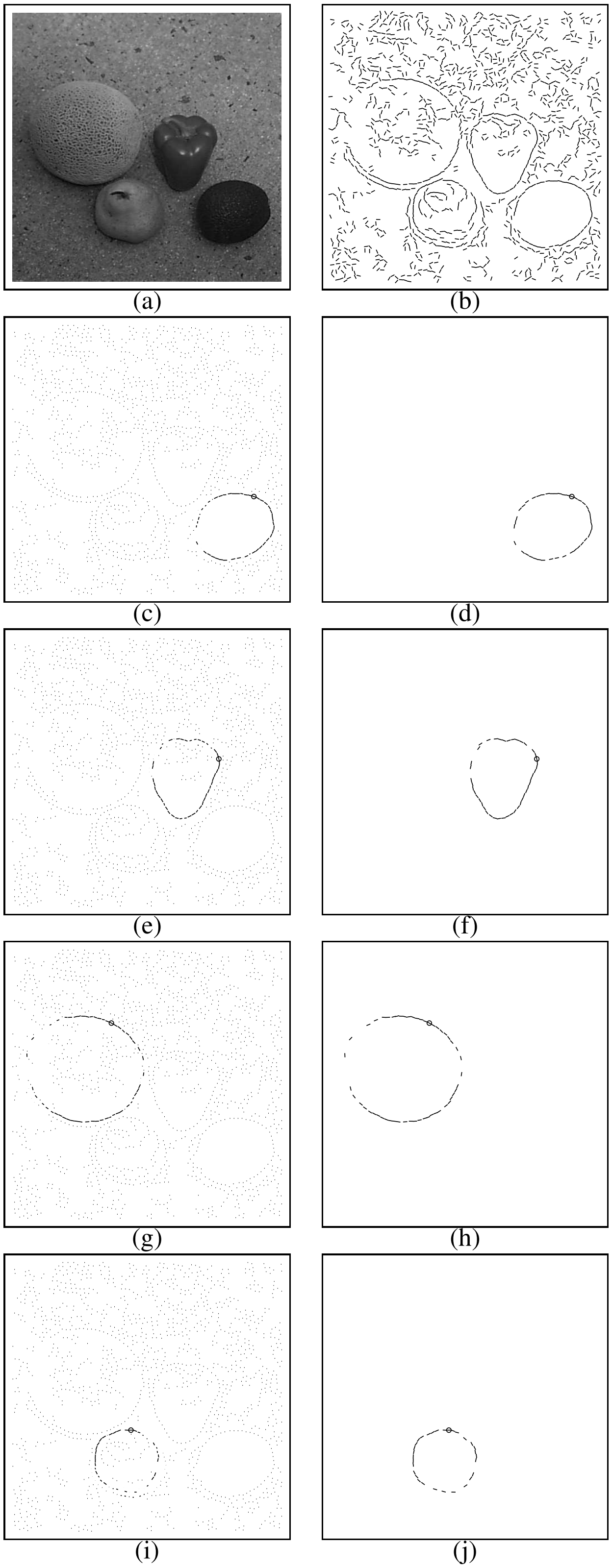
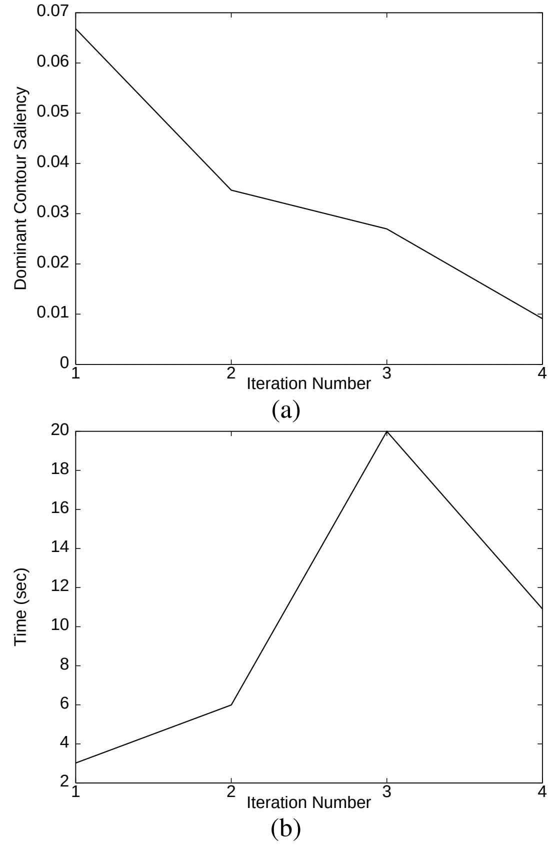


Fig. 4. Fruits on concrete. (a) Greyscale image. (b) Canny edge output. (c)-(j) First four iterations of the segmentation algorithm. For each iteration the saliencies are shown on the left and the segmentation is shown on the right. In each saliency plot, the length of an edge is propotional to the saliency of that edge. The most salient edge in both the saliency and segmentation plots are shown inside the small circle.

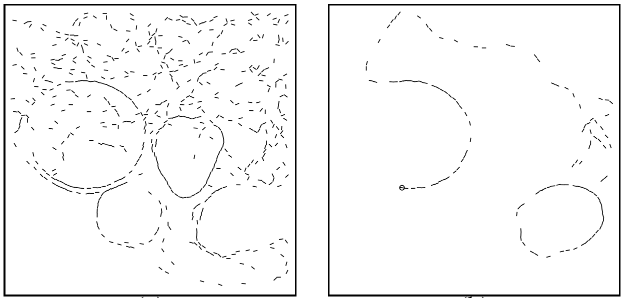


least time of ≈ 3 sec since the contour extracted is relatively salient (as seen from the dominant contour saliency plot above) compared to the other contours in the scene. The third iteration took the longest time of ≈ 20 sec– possibly because of the large gaps in the contour being extracted (bounding the cantelope). The average time for all iterations is ≈ 9.9 sec.

B. Importance of Global Information

In this section, we will show the importance for segmentation of the global information encoded by the link saliency matrix $\mathbf{C}$ by replacing it with the affinity matrix $\mathbf{P}$ which encodes only local information. The edge saliency vector $\mathbf{c}$ is left unchanged. With this replacement, the segmentation algorithm extracts out the same contour in the first iteration as the algorithm using 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. 5 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.

In a previous paper[25], a purely local saliency measure (termed WJ) was judged to be more effective in isolating smooth closed contours in the presence of background clutter than three other well known saliency measures (those termed GM, SB, and SU and based on [8], [17], and [20]). Analogous to the expression, $c_i = s_i \bar{s}_i$ (where $\mathbf{s}$ and $\bar{\mathbf{s}}$ are right and left eigenvectors with largest positive real eigenvalue of $\mathbf{P}$) the saliency of an edge according to the WJ measure is $w_i = x_i \bar{x}_i$, where $x_i = \sum_j P_{ij}$. We observe that the WJ measure can be seen as a single step of the power-method iteration necessary for



(a) (b)

Fig. 5. Fruits on concrete. (a) When the global link saliency matrix, **C**, is replaced by the local affinity matrix, **P**, from which it is derived, the segmentation algorithm fails in the second iteration. (b) When the global link saliency matrix, **C**, is replaced by a local link saliency matrix, **W**, based on the WJ saliency measure, the segmentation algorithm fails in the first iteration.

computing the eigenvector with largest positive real eigenvalue of **P**. It follows that by comparing the performance of a segmentation algorithm based on the WT measure to one based on the WJ measure, we can ascertain the value of power-method iterations beyond the initial step. This speaks directly to the important issue of iterative versus non-iterative (i.e., voting) methods in perceptual organization, an issue which is explored extensively in [13].

Using reversal symmetry, $P_{ij} = P_{\overline{j}\overline{i}}$, we can easily show that $w_i = \sum_{j,k} P_{ki} P_{ij}$ which implies that the saliency for edge i is the sum of the probabilities of contours of length two centered on i. Because the saliency is determined solely by the probabilities of length two contours, it follows that the global property of contour closure plays no tole in determining edge saliency. Consequently, edges forming a closed contour can be of low saliency despite the fact that they contain many closed contours of relatively high probability.

In our second demonstration of the importance of using global information, the global link saliency matrix, **C**, based on the WT measure, is replaced by a local link saliency matrix, **W**, based on the WJ measure. The expression for local link saliency is analogous to the expression for global link saliency, Eq. 5. Specifically,

$$W_{ij} = \bar{x}_i P_{ij} x_j. \tag{12}$$

Using reversal-symmetry, it can be shown that

$$W_{ij} = \sum_{k,l} P_{ki} P_{ij} P_{jl}. \tag{13}$$

Thus the W_{ij}'s are proportional to the probability that a contour of length three is centered on the link $j \rightarrow i$. However, when we use these local measures, the segmentation algorithm breaks down in the first iteration as shown in Fig. 5 (b). The most salient edge according to the w_i's (indicated by the circle) lies on the cantelope, which was the third fruit extracted using the global saliency measure. While tracing the contour bounding the cantelope, the algorithm loses its way when it encounters the large gaps. In summary, the c_i's are essential for reliably determining the starting edge for the segmentation algorithm, and the C_{ij}'s are essential for bridging large gaps. Both are functions of the eigenvector with largest positive real eigenvalue of the **P** matrix.

C. Additional Segmentation Examples

The eigen-solver for the matrix **P** described above (see [23] 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

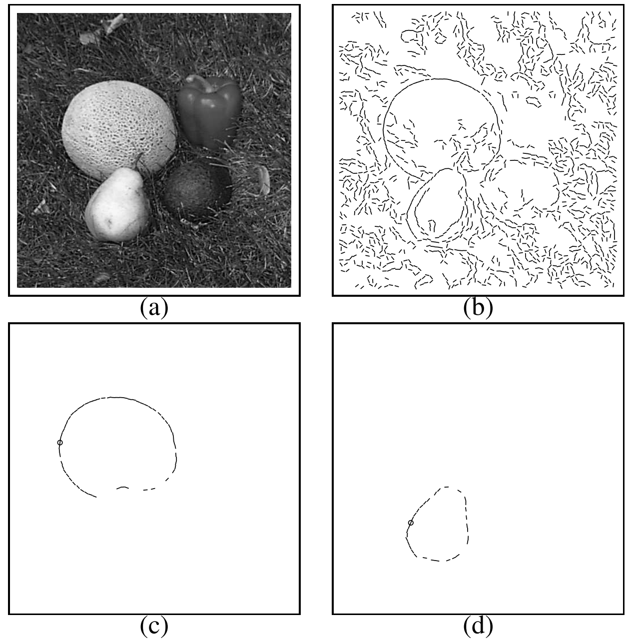


Fig. 6. Fruits on grass. (a) Greyscale image (b) Canny edge output (c)-(d) first and second segmentations.

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. 6 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.

Fig. 7(a) shows an example where there are significant shadows which produce strong smooth contours adjacent to the stones. However, since they are not closed, the shadow contours are not as salient as the contours which actually bound the stones. Consequently, they do not confuse the algorithm.

Finally, Fig. 8(a) is an image of a large number of coins on a tabletop. Although this is an image which would be relatively easy to segment using image brightness, the segmentation which is shown in Fig. 8(c) has been computed solely using the Canny edges shown in Fig. 8(b).

VI. Finding the Optimum Speed

The shape distribution which is used to build the **P** matrix is defined by three parameters, T, τ, and γ. Although there has been some interesting recent work on learning parameter settings for grouping algorithms (see [18]), we have simply selected values for T, τ, and γ which we have found to yield good results in practice. In this section, we describe preliminary work on choosing the value of one of these parameters, γ, the particle's speed, automatically.

The segmentation algorithm which we have described assumes a fixed value for γ. However, there are two properties

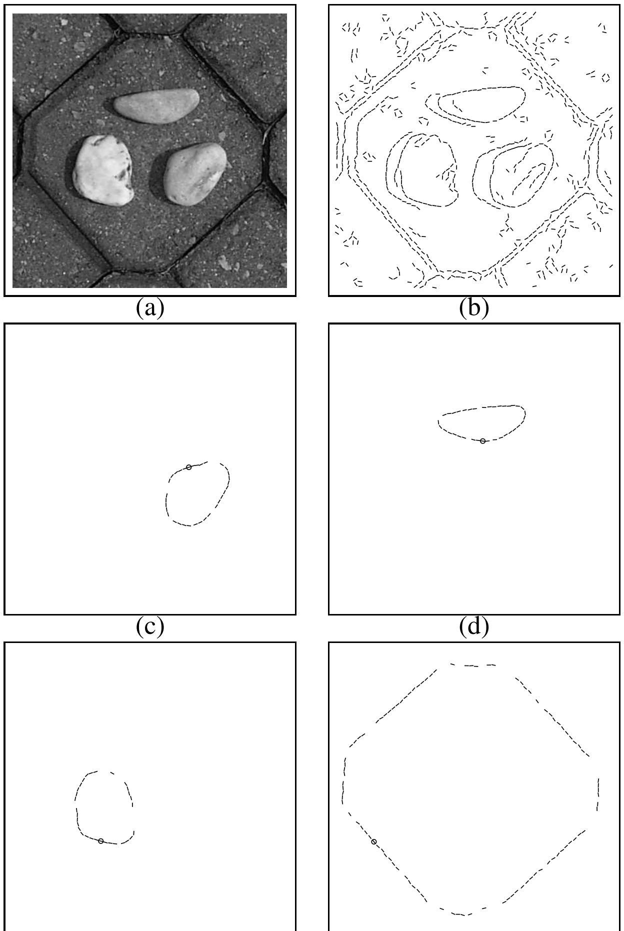


Fig. 7. Stone on pavement. (a) Greyscale image (b) Canny edge output (c)-(f) first four segments.

of the shape distribution which are directly affected by the particle's speed. First, the distance a particle travels before it decays *increases* with increasing speed. Second, the variance in a particle's direction of motion relative to the distance it travels *decreases* with increasing speed. Consequently, the choice of γ effectively determines both the curvature of the closed contours which will be classified as most salient, and the optimum distance between adjacent edges. A more principled approach would be to isolate closed contours irrespective of their average curvature and irrespective of the average distance between adjacent edges. One way to do this would be to systematically vary speed within Algorithm 1 so that the contour which is most salient, *i.e.,* the contour with maximum eigenvalue, among contours of all possible γ is extracted. In principle, the saliency of contours of different average curvature and different edge sampling rates would be maximized at different speeds, resulting in a more robust segmentation algorithm. See Algorithm 4.

Algorithm 4: Extract an object (scale-invariant).

$extract(A)$
begin
$\gamma_{\max} \leftarrow \underset{\gamma}{\operatorname{argmax}}\ eigenvalue(affinities(A, \gamma))$
$\mathbf{s} \leftarrow eigenvector(affinities(A, \gamma_{max}))$
$j = \underset{i}{\operatorname{argmax}}\ (\bar{s}_i\ s_i)$
return $reachable(j) \cap \overline{reachable}(\bar{j})$
end

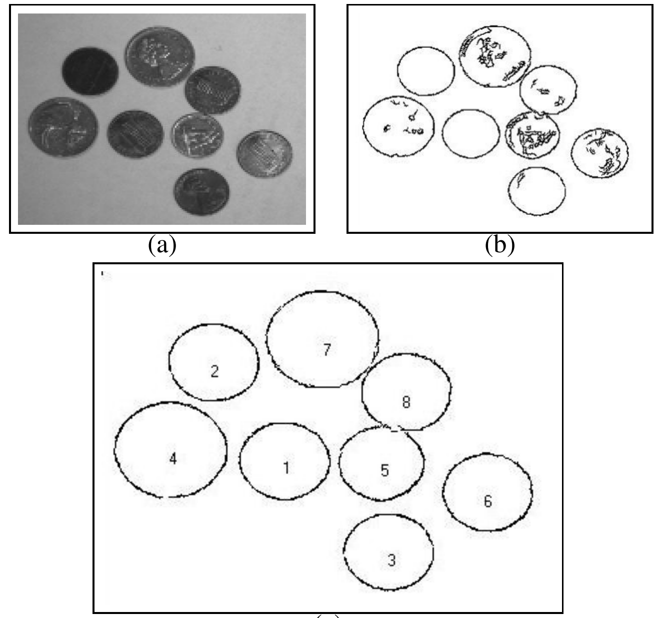


Fig. 8. *Coins*. (a) Original image. (b) Edge input obtained by Canny detector. (c) Segmented objects, numbered in the order in which they are extracted.

Fig. 9(a) is an image of three fruits on a wooden table with prominent wood grain background. The segmentation shown in Fig. 9(c) was computed using Algorithm 4 instead of Algorithm 1. The optimal speed within the range $[0.01, 1.5]$ for each object was computed using Brent's method[3], which does not require analytic derivatives, and is able to locate a local optimum in $\lambda(\gamma)$.

In our initial attempt to run the modified segmentation algorithm on this image, we used the same values for T and τ used to compute the other results in this paper. Unfortunately, the γ which maximized the eigenvalue was 1.5. Because this value is on the boundary of the search interval, it is not actually a local optimum of $\lambda(\gamma)$, and the resulting segmentation was of very low quality. After increasing the diffusion constant, T, from 0.004 to 0.007 and the decay constant, T, from 5.0 to 6.5, the optimization procedure returned values of γ which were within the range $[0.01, 1.5]$, which implies that they are true local optima. The optimal speed, γ_{max}, for the first object was 1.23 and 1.27 for the second. The eigenvalue, $\lambda(\gamma_{max})$, at the optimal speed was 0.0099 for the first object and 0.0071 for the second. The closed contours are of good quality. See Fig. 9(c). The algorithm failed to find the third object, because the Canny edge detector returned very few edges which lie on its boundary.

VII. Conclusion

We have demonstrated how an edge saliency measure based on the global property of contour closure can form the basis of a segmentation algorithm able to identify multiple salient closed contours in real images. More specifically, we have demonstrated that computing the connected-components of a graph based on a matrix which makes explicit the relative number of closed contours containing pairs of edges, is more effective than searching a graph based on local information about pairs of edges in isolation.

Our approach to grouping edges into salient closed contours involves the solution of an eigenvector/eigenvalue problem. Recently, other researchers[15], [17], [18], [21] have also proposed grouping image features by solving eigenvector/eigenvalue problems.

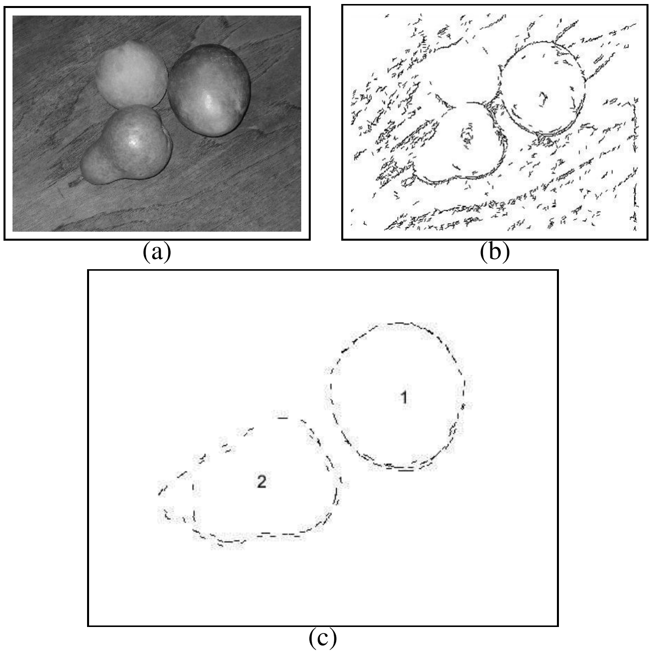


Fig. 9. *Fruits*. (a) Original image. (b) Edge input obtained by Canny detector. (c) Segmented objects, numbered in the order in which they are extracted. The optimal speed, γ_{max}, for the first object was 1.23 and 1.27 for the second. The eigenvalue, $\lambda(\gamma_{max})$, at the optimal speed was 0.0099 for the first object and 0.0071 for the second.

The normalized min-cut approach described in [21] can group more general image features than our approach can. However, since we restrict ourselves to grouping edges, we are able to impose the important constraint of tangent continuity, which has no counterpart for non-edge features such as texture or brightness. Furthermore, any approach enforcing tangent continuity using the mechanism of edge-directionality requires a non-symmetric affinity matrix $\mathbf{P}$ for which the min-cut approach proposed in [21] does not apply. As previously noted, the use of a symmetric affinity matrix makes contours containing cusps salient (see the discussion in Section II). Hence, we would expect poor performance on edge grouping problems with a min-cut approach since tangent continuity cannot be enforced.

Like [21], the dominant eigenvector based method described by [17] is applicable to features other than edges. Also like [21], this method assumes that the affinity matrix is symmetric and therefore cannot use edge-directionality to enforce tangent continuity.

Of course, the generic grouping algorithms of [1], [7], [15], [21] make much weaker assumptions about the input image than does the algorithm which we describe here. The algorithm we describe is specifically designed to group edges into smooth closed contours. When an image does not contain closed contours, when the contours it contains are not smooth, or when local edge detection processes fail because of lack of contrast, *i.e.*, when our assumptions are violated, our method will fail. It is possible that in such cases, generic grouping methods, which are able to organize a wider variety of image features, and which make weaker assumptions about them, may succeed in such cases.

However, we believe that grouping methods which are designed to solve a specific grouping problem, such as grouping edges into smooth closed contours, will outperform general-purpose methods on images for which their assumptions hold. This is because generic methods cannot fully exploit domain specific contraints such as contour closure and tangent continuity, which have no counterparts for non-edge features such as texture or brightness.

Appendix A

First we prove some preliminary lemmas.

Lemma 1: If $\mathbf{s}$ is a (right) eigenvector of $\mathbf{P}$ with eigenvalue λ then $\bar{\mathbf{s}}$ is a left eigenvector of $\mathbf{P}$ or equivalently a (right) eigenvector of P^{T} with the same eigenvalue.

Proof. Taking the i-th component of $\mathbf{P}^{\mathrm{T}}\bar{\mathbf{s}}$,

$$\sum_j P_{ji}\bar{s}_j = \sum_j P_{\bar{i}\bar{j}} s_{\bar{j}} \tag{14}$$

$$= \sum_j P_{\bar{i}j} s_j \tag{15}$$

$$= \lambda \bar{s}_i. \tag{16}$$

Hence, $\bar{\mathbf{s}}$ is an eigenvector for $\mathbf{P}^{\mathrm{T}}$ or equivalently a left eigenvector for $\mathbf{P}$ with the same eigenvalue λ.

Lemma 2: For an irreducible positive matrix $\mathbf{P}$ that is reversal-symmetric

$$\lim_{n\to\infty}\left(\frac{\mathbf{P}}{\lambda}\right)^n = \mathbf{s}\cdot\bar{\mathbf{s}}^{\mathrm{T}} \tag{17}$$

where λ is the largest eigenvalue of $\mathbf{P}$ and $\mathbf{s}$ is the corresponding eigenvector.

Proof. For a general irreducible positive matrix $\mathbf{A}$ it is shown in[10] that

$$\lim_{n\to\infty}\left(\frac{\mathbf{A}}{\lambda}\right)^n = \mathbf{x}\cdot\mathbf{y}^{\mathrm{T}} \tag{18}$$

where $\mathbf{x}$ and $\mathbf{y}$ are respectively the right and left eigenvectors of $\mathbf{A}$ corresponding to the largest eigenvector λ normalized such that $x^{\mathrm{T}}y = 1$. From the previous lemma, we know that if $\mathbf{s}$ is a left eigenvector of a reversal-symmetric matrix $\mathbf{P}$, then $\bar{\mathbf{s}}$ is the corresponding right eigenvector with the same eigenvalue. Hence the proof.

Proof of the First Saliency Theorem (Theorem 1). First we note the simple relationship between the diagonal elements of the powers of the affinity matrix $\mathbf{P}$ and probabilities of closed contours. $(P^k)_{ii}$ is the sum of the probabilities of all closed contours of length k that pass through edge i. Using this relationship and letting λ be the largest eigenvalue of $\mathbf{P}$, the definition for c_i in Equation 2 can be rewritten in terms of the powers of $\mathbf{P}$ as :

$$c_i = \lim_{n\to\infty}\frac{(\mathbf{P}^n)_{ii}}{\sum_j(\mathbf{P}^n)_{jj}} \tag{19}$$

$$= \lim_{n\to\infty}\frac{(\frac{\mathbf{P}}{\lambda})^n_{ii}}{\sum_j(\frac{\mathbf{P}}{\lambda})^n_{jj}}. \tag{20}$$

The above limit exists if the limit for both the numerator and the denominator exists and the limit for the denominator is non-zero. Using Lemma 2 :

$$\lim_{n\to\infty}\left(\frac{\mathbf{P}}{\lambda}\right)^n_{ii} = s_i\bar{s}_i \tag{21}$$

where the s_i's are the components of the eigenvector of $\mathbf{P}$ corresponding to its largest eigenvalue λ and assuming that the eigenvector is normalized such that $\sum_j s_j\bar{s}_j = 1$. Hence, both the limits for the numerator and denominator in the ratio (20) exist and is equal to respectively, $s_i\bar{s}_i$ and $\sum_j s_j\bar{s}_j$. Finally, we note that the limit for the denominator in the ratio is non-zero. For our problem there is a non-zero probability that a contour passes through any two edges in the image in succession[1]. Hence **P** is positive[10] and according to *Perron*'s theorem [10] for positive matrices, all the components of the eigenvector corresponding to the largest eigenvalue are positive. Hence, $\sum_j s_j \bar{s}_j > 0$ and hence the limit of the denominator in the ratio (20) is non-zero. Since we assume that the eigenvector is normalized so that $\sum_j s_j \bar{s}_j = 1$, the expression for c_i becomes :

$$\begin{aligned} c_i &= \lim_{n\to\infty} \frac{(\frac{\mathbf{P}}{\lambda})^n_{ii}}{\sum_j (\frac{\mathbf{P}}{\lambda})^n_{jj}} && (22)\\ &= \frac{\lim_{n\to\infty}(\frac{\mathbf{P}}{\lambda})^n_{ii}}{\lim_{n\to\infty}\sum_j (\frac{\mathbf{P}}{\lambda})^k_{jj}} && (23)\\ &= \frac{s_i\bar{s}_i}{\sum_j s_j\bar{s}_j} && (24)\\ &= s_i\bar{s}_i. && (25)\end{aligned}$$

Proof of the Second Saliency Theorem (Theorem 2). The probability that closed contours of length n pass through edges j and i successively is given by $P^{n-1}_{ji}P_{ij}$ since all such contours pass through the link from edge j to edge i at least once. Hence we can rewrite the definition for the link saliencies (4) as :

$$C_{ij} = \lim_{n\to\infty} \frac{\left(\frac{\mathbf{P}}{\lambda}\right)^{n-1}_{ji}\cdot\left(\frac{P_{ij}}{\lambda}\right)}{\sum_l (\frac{\mathbf{P}}{\lambda})^n_{ll}} \quad (26)$$

Again, using the limit theorem in Lemma (2) and arguments similar to that made in the proof of Theorem (1) on the existence of limits, we have :

$$\begin{aligned} C_{ij} &= \frac{\lim_{n\to\infty}\left(\frac{\mathbf{P}}{\lambda}\right)^{n-1}_{ji}\cdot\left(\frac{P_{ij}}{\lambda}\right)}{\sum_l \lim_{n\to\infty}(\frac{\mathbf{P}}{\lambda})^n_{ll}} && (27)\\ &= \frac{s_j\bar{s}_i\left(\frac{P_{ij}}{\lambda}\right)}{\sum_l s_l\bar{s}_l} && (28)\\ &= \frac{\bar{s}_iP_{ij}s_j}{\lambda}. && (29)\end{aligned}$$

Theorem 3—Third Saliency Theorem: Given a set $\mathcal{C}$ of closed contours in an image, consider the induced graph G whose vertices are edges from the image. The only links between vertices correspond to the directed links between successive edges of the contours in $\mathcal{C}$. Then G is partitioned into isolated strongly-connected components, no two of which have any link between them.

Proof. It is easily seen that G has isolated strongly-connected components iff for two edges i and j, there is a path $i \rightsquigarrow j$ iff there is a path $j \rightsquigarrow i$. Hence in our case, we need to prove that the above condition between two edges i and j always holds. It is enough to show this for simple paths where there are no loops. Any path $i \rightsquigarrow j$ can be decomposed into a sequence of subpaths each of which is a subsequence fully contained in some closed contour of $\mathcal{C}$. The subsequences are constructed in the following manner. Consider the set A_i of all the contours in $\mathcal{C}$ that contain edge i. Starting from edge i we trace out the the path $i \rightsquigarrow j$. As we move along this path, we remove from the set A_i any closed contours that does not contain the whole subsequence seen so far. Then either we reach edge j before exhausting the contours in A_i or A_i becomes empty at some intermediate edge. In the former case, any of the remaining closed contour in A_i provides a return path to edge i from j by tracing out the rest of such a closed contour. In the latter case, let k be the last intermediate edge after which the set A_i becomes empty. The path from $i \rightsquigarrow k$ is the first subsequence that we construct. At k there still exists some closed contour in A_i. Thus there is a return path $k \rightsquigarrow i$ by completing any such closed contour remaining in A_i. We recursively construct the remaining subpaths by considering the path $k \rightsquigarrow j$ and starting with the set A_k which is the set of all contours in $\mathcal{C}$ that pass through edge k. As argued above, each such subpath has a returning path from the end of the subpath to its beginning. Hence all the returning paths can be concatenated together in reverse order to get a returning path from $j \rightsquigarrow i$. Hence $i \rightsquigarrow j$ iff $j \rightsquigarrow i$. A strongly connected component of a graph is defined [5] as a subset of nodes where for any two nodes i and j in the subset, there exists a path from $i \rightsquigarrow j$ and from $j \rightsquigarrow i$. Hence in our case since $i \rightsquigarrow j$ iff $j \rightsquigarrow i$, if there exists any path $i \rightsquigarrow j$, then edges i and j belong to some strongly connected component. Hence the whole graph can be partitioned into a set of isolated strongly connected components with no links between any two of the components.

[1] This is not necessarily true when we consider sparse representations of the matrix **P**, but for such a case all that is required is that there is a non-zero probability that a contour start from edge i and end in edge j with the possibility of threading through intermediate edges.

Appendix B

In this appendix, we give an analytic expression characterizing the probability distribution of boundary-completion shapes derived in [22]. For a derivation of a related affinity function see [19]. We define the affinity, P_{ji}, between two directed edges, i and j, to be

$$P(j \mid i) = \int_0^\infty P(j \mid i; t)\, dt \;\approx\; FP(j \mid i; t_{opt}) \quad (30)$$

where $P(j \mid i; t)$ is the probability that a particle which begins its stochastic motion at (x_i, x_i, θ_i) at time *0* will be at (x_j, y_j, θ_j) at time *t*. This probability is defined to be the sum over the probabilities of all paths that a particle can take between the two edges. This integral is approximated analytically. The approximation is the product of P evaluated at the time at which the integral is maximized, *i.e.*, t_{opt}, and a weighting factor, *F*. The expression for P at time *t* is

$$P(j \mid i; t) = \frac{3\,\exp[-\frac{6}{Tt^3}(at^2 - bt + c)]\,\exp(-\frac{t}{\tau})}{\sqrt{\frac{\pi^3T^3t^7}{2}}} \quad (31)$$

where

$$\begin{aligned} a &= \frac{2 + \cos(\theta_j - \theta_i)}{3} && (32)\\ b &= \frac{x_{ji}(\cos\theta_j + \cos\theta_i) + y_{ji}(\sin\theta_j + \sin\theta_i)}{\gamma} && (33)\\ c &= \frac{(x_{ji}^2 + y_{ji}^2)}{\gamma^2} && (34)\end{aligned}$$

for $x_{ji} = x_j - x_i$ and $y_{ji} = y_j - y_i$. The distribution of shapes is determined by the half-life, τ, the variance, *T*, and the speed of the particle, γ. The expression for P should be evaluated at $t = t_{opt}$, where t_{opt} is real, positive, and satisfies the cubic equation

$$-\frac{7t^3}{4} + \frac{3(at^2 - 2bt + 3c)}{T} = 0. \quad (35)$$

If more than one real, positive root exists, then the root maximizing $P(j \mid i; t)$ is chosen. Finally, the extra factor F is

$$F = \sqrt{\frac{2\pi t_{opt}^5}{\frac{12(3c - bt_{opt})}{T} + \frac{7t_{opt}^3}{2}}}. \tag{36}$$

Acknowledgements

It is a pleasure to thank Satish Rao for first suggesting a connection between our segmentation algorithm and strongly-connected components. SM and LW were partially supported by NEC Research Institute. LW and KX were partially supported by Los Alamos National Laboratory.

References

[1] Amir, A. and Lindenbaum, M. 1998. A generic grouping algorithm and its quantitative analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence* **20**(2):168-185.

[2] Boldt, M., Weiss, R. and Riseman, E. 1989. Token-based extraction of straight lines. *IEEE Trans. on Systems, Man and Cybernetics* **19**(6):1581–1594.

[3] Brent, R.P., 1973. *Algorithms for Minimization without Derivatives*, Prentice-Hall.

[4] Canny, J. 1986. A computational approach to edge detection, *IEEE Trans. Pattern Analysis and Machine Intelligence* **9**(6):679–698.

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

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

[7] Gdalyahu, Y. Weinshall, D. and Werman, M. 1999. A Randomized Algorithm for Pairwise Clustering, *Advances in Neural Information Processing Systems***11**, Denver, CO.

[8] Guy, G., and Medioni, G. 1996. Inferring Global Perceptual Contours from Local Features, *Intl. Journal of Computer Vision* **20**:113–133.

[9] Herault, L. and Horaud, R. 1993. Figure-ground discrimination: A combinatorial optimization approach, *IEEE Trans. on Pattern Analysis and Machine Intelligence* **15**:899-914.

[10] Horn, R.A. and Johnson, C.R. 1990. *Matrix Analysis*, Cambridge University Press, Cambridge, UK. Chapter 8.

[11] Jacobs, D. 1993. Robust and Efficient Detection of Convex Groups. In *CVPR '93*, New York, NY.

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

[13] Medioni, G., Lee, M. and Tang, C. 2000. *A Computational Framework for Segmentation and Grouping*, Elsevier Science B.V.

[14] Mumford, D. 1994. Elastica and computer vision, In *Algebraic Geometry and Its Applications*, Chandrajit Bajaj (Ed.), Springer-Verlag, New York.

[15] Perona, P. and Freeman, W. 1998. A Factorization Approach to Grouping. In *ECCV '98*, Freiburg, Germany. **Vol. I**:655–670.

[16] Raman, S.V., Sarkar, S. and Boyer, K.L. 1993. Hypothesizing Structures in Edge-focused Cerebral Magnetic Resonance Images Using Graph-theoretic Cycle Enumeration, *Computer Vision, Graphics and Image Processing* **57**(1):81-98.

[17] Sarkar, S. and Boyer, K. 1998. Quantitative Measures for Change based on Feature Organization: Eigenvalues and Eigenvectors, *Computer Vision and Image Understanding* **71**:110–136.

[18] Sarkar, S. and Soundararajan, P., 2000. Supervised Learning of Large Perceptual Organization : Graph Spectral Partitioning and Learning Automata, *IEEE Trans. Pattern Analysis and Machine Intelligence* **22**(5).

[19] Sharon, E., Brandt, A., and Basri, R. 1997. Completion Energies and Scale, In *CVPR '97*, Puerto Rico, USA.

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

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

[22] Thornber, K.K. and Williams, L.R. 1996. Analytic Solution of Stochastic Completion Fields. *Biological Cybernetics* **75**:141–151.

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

[24] Williams, L.R. and Jacobs, D.W. 1997. Stochastic Completion Fields: A Neural Model of Illusory Contour Shape and Salience. *Neural Computation* **9**(4):837–858.

[25] Williams, L.R. and Thornber, K.K. 2000. A Comparison of Measures for Detecting Natural Shapes in Cluttered Backgrounds. *Intl. Journal of Computer Vision* **34**(2/3): 81–96.

[26] Williams, L.R. and Thornber, K.K. 2001. Orientation, Scale and Discontinuity as Emergent Properties of Illusory Contour Shape, *Neural Computation* **13**(8):1683-1711.