# Properties of the Affinity Matrix for Multiple Closed Contour Segmentation

**Kanglin Xu and George F. Luger**

Department of Computer Science
University of New Mexico
Albuquerque, NM 87131
{klxu, luger}@cs.unm.edu

*Abstract*—**In this paper, we describe an algorithm based on a distribution of contours traced by particle moving in directions undergoing Brownian motion for a segmentation problem. We analyze the properties of the affinity matrix, whose elements are the edge-to-edge transition probabilities, from a mathematical point of view and give formal proofs. This analysis is essential because we can use the properties of the affinity matrix to reduce the amount of computation greatly when solving segmentation problems, especially for large real images. The results obtained can be used for robot visual servo to pick multiple objects in the scene.**

## I. Introduction

Many objects in the physical world are bounded by smooth closed contours. Human visual system is able to perceive the bounding contours of objects even in the absence of contrast due to occlusion. In computer vision, we attempt to emulate the way the human visual system behaves. Given an input image, we can segment different objects in it based on a computational model. The purpose of segmentation is to partition a set of edges into subsets so that the edges in different subsets bound different objects in the scene. In this paper, we call this the *segmentation problem*.

The segmentation problem has been and still is an important topic in the field of computer vision. Basically the previous approaches can be divided into the following categories [5]:

1) Histogram based segmentation
2) Neighborhood based segmentation
3) Surface fitting based segmentation
4) Physically based segmentation

Among these approaches, many of them belong to the neighborhood category. They are based on the properties of pixels which are used to related to their local neighborhood [4]. The *grouping problem*, or the *saliency problem* mentioned in [14] is also in the second category. In this paper, we will focus on the *saliency problem*.

Many existing approaches (e.g. [7], [8], [15]) have used graph-based search techniques to find closed contours of different objects. For the *saliency problem*, the entries of the affinity matrix, from which the graph is built, are the probabilities that a smooth contour passes through a pair of edges. The affinity values are based on the Gestalt principles of proximity and smooth-continuation and derive from the local properties of position and orientation of the two edges. Mahamud, Williams, Thornber and Xu[7] show that the effectiveness of these approaches is limited. In addition to the local properties, i.e. proximity and smooth-continuation, Mahamud *et al.* [7] use an approach based on global saliency computation. They use the local affinity measure to compute a global saliency measure. Concretely, they first compute the affinity matrix, **P**, and the eigenvector associated with its maximum eigenvalue, and then compute a link saliency matrix, **C**. Only after the computation of the global saliency measure, do they use a depth-first-search to find the strongly connected components [7].

Like the other algorithms for solving the segmentation problem, Mahamud's algorithm requires a large number of floating point operations. Williams and Thornber [14] first use the properties of the affinity matrix to reduce the amount of computation greatly when they try to find the eigenvalue and eigenvector of the affinity matrix. Unfortunately, although previous papers [7], [14] mention and use this idea, none of them give a detailed analysis, formal proof, and an explicit approach. This is what our paper is to accomplish. In addition, we first present the symmetric properties of the affinity matrix's submatrices, which have not been introduced before. We can reduce computation further by using them when we build an affinity matrix. We include this analysis here as complement.

The remainder of this paper is organized as follows: Section 2 analyzes and proves the affinity matrix properties. Section 3 describes the segmentation algorithm. Section 4 gives experimental results, and Section 5 offers concluding remarks.

## II. Saliency Measures and Affinity Matrix Properties

In this section, we define the affinity matrix, the saliency measure used and present its properties. We then give the formal proof.

### A. Edge-to-edge Transition Probability

Suppose we have an input consisting of $N$ edges. $\vec{x}_i$ is the location and $\theta_i$ is the orientation of edge $i$. Now we add another $N$ edges. For each edge $i$, we add edge $\bar{\imath}$ and let its location be the same as that of $i$ but its orientation be equal to $i$'s orientation plus $\pi$, that is, $\vec{x}_i = \vec{x}_{\bar{\imath}}$ and $\theta_i = \theta_{\bar{\imath}} + \pi$. This step is necessary because it ensures that a contour arriving in a given direction must exit in the same direction. Basically, it ensures that tangent continuity is satisfied. Without this

Kanglin Xu and George Luger. "Properties of the Affinity Matrix for Multiple Closed Contour Segmentation." In Proceedings of the 12th International IEEE Conference on Advanced Robotics, July 18-20, 2005, Seattle, Washington.

mechanism, it is possible to obtain a contour containing *cusps* (i.e., reversals in direction). Williams and Thornber[14] give a more detailed explanation.

Given two directed edges, *i* and *j*, Williams *et al.* [14] derive an expression for the probability, $P(j|i)$, that a particle moving with constant speed in a direction given by a Brownian motion will travel from edge *i* to edge *j* and visit $n-1$ intermediate edges. From [14], we know that $P(j|i)$ is dependent on three parameters in addition to the locations and orientations of edges *i* and *j*: (1) the half-life $\tau$, (2) the variance *T* of the directional change of the particle, and (3) the speed of the particle, $\gamma$, and the locations and orientations of edges *i* and *j*. The half-life, $\tau$, models the principle proximity while the variance, *T*, models smooth-continuation. These three parameters determine the distribution of shapes.

### B. Edge Saliency

In this paper, the saliency of edge *i* is defined to be the relative number of closed contours that pass through this edge. If we assume that the closed contour begins and ends at edge *i*, we can write the definition of saliency as follows

$$c_i = \lim_{t \to \infty} \frac{(\mathbf{P})^t_{ii}}{\sum_j (\mathbf{P})^t_{jj}}$$

where $\mathbf{P}$ is the affinity matrix whose elements are the edge-to-edge transition probabilities given by $P(j|i)$. Williams and Thornber [14] have shown that the edge saliency based on the above definition can be evaluated as

$$c_i = \frac{x_i \bar{x}_i}{\sum_j x_j \bar{x}_j}$$

where $\mathbf{x}$ is the right eigenvector corresponding to the largest eigenvalue of the affinity matrix, $\mathbf{P}$, and $\bar{\mathbf{x}}$ is the corresponding left eigenvector. If we do not consider the constant factor, $\sum_j x_j \bar{x}_j$, or normalize the eigenvector such that $\sum_j x_j \bar{x}_j = 1$, we have $c_i = x_i \bar{x}_i$.

### C. Properties of the Affinity Matrix

As mentioned before, the affinity matrix is defined by edge-to-edge transition probabilities given by $P(j|i)$. In this section, we will analyze and prove its properties.

*Lemma 1: If edge i and edge $\bar{\imath}$ have an identical location but opposite directions, the affinity matrix* $\mathbf{P}$ *which is created using the equations described by Williams and Thornber[14] has the following properties:*

1) $P_{ij} = P_{\bar{j}\bar{\imath}}$. *This is termed reversal symmetry.*
2) *In the above equation, we can substitute i for $\bar{\imath}$ everywhere, and $\bar{\imath}$ for i everywhere.*
3) $P_{\bar{\imath}i} = P_{i\bar{\imath}}$ *is constant, and* $P_{ii} = P_{\bar{\imath}\bar{\imath}}$ *is constant (for all* $i = 1...n$*).*

*Proof:* As mentioned before, the affinity, $P_{ji}$, is dependent on three parameters as well as the locations and orientations of edges *i* and *j*. If we focus on these three parameters, from the expression given by Williams and Thornber[14] we know that the affinity, $P_{ji}$, is determined by three factors related to the locations and orientations of edges *i* and *j*. Let these three factors be *a, b, c*:

$$\begin{aligned}
a &= \frac{2 + \cos(\theta_j - \theta_i)}{3} \\
b &= \frac{x_{ji}(\cos\theta_j + \cos\theta_i) + y_{ji}(\sin\theta_j + \sin\theta_i)}{\gamma} \\
c &= \frac{(x_{ji}^2 + y_{ji}^2)}{\gamma^2} \\
x_{ji} &= x_j - x_i \\
y_{ji} &= y_j - y_i.
\end{aligned}$$

Given the above equations, the proof is very straightforward. We just substitute $x_i = x_{\bar{\imath}}$, $y_i = y_{\bar{\imath}}$, and $\theta_{\bar{\imath}} = \theta_i + \pi$ into the above three equations. In this way, we can prove $P_{ij} = P_{\bar{j}\bar{\imath}}$, $P_{\bar{\imath}j} = P_{\bar{j}i}$ and $P_{i\bar{j}} = P_{j\bar{\imath}}$. Finally, since $\theta_i = \theta_{\bar{\imath}} + \pi$, $a_{ii}$ and $b_{ii}$ are constant. Therefore, Property *3* is proved. Note that the *reversal symmetry* property means that the probability that any particle travels along a curve starting from edge, *j*, and ending in edge, *i*, is the same as that of a particle traveling from edge, $\bar{\imath}$, to edge, $\bar{j}$. ■

Consider an input pattern consisting of *N* edges. We construct a set of $2N$ states, *S*. For each edge, *i*, in the input pattern, there are two states, *i* and $\bar{\imath}$, in *S*. If we arrange the set, *S*, so that the order of the first half of the $2N$ elements is the same as the input order (i.e., edge *i's*) and the second *N* of the *2N* elements are edges with opposite directions to each element in the first half (i.e., edge $\bar{\imath}$*'s*), we obtain the following lemma:

*Lemma 2: Given an input vector* $A[1..2N]$ *as described above, if we use the following loops*

```
$for(i = 1; i \leq 2N; i++)$
    $for(j = 1; j \leq 2N; j++)$
        $P(i|j) = function(A[i], A[j])$
```

*to create the affinity matrix, we obtain* $\mathbf{P}$ *as follows:*

$$\mathbf{P} = \begin{bmatrix} \{P_{ij}\} & \{P_{i\bar{j}}\} \\ \{P_{\bar{\imath}j}\} & \{P_{\bar{\imath}\bar{j}}\} \end{bmatrix}$$

*where*

$$\{P_{ij}\} = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1N} \\ P_{21} & P_{22} & \cdots & P_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1} & P_{N2} & \cdots & P_{NN} \end{bmatrix}$$

$$\{P_{i\bar{j}}\} = \begin{bmatrix} P_{1\bar{1}} & P_{1\bar{2}} & \cdots & P_{1\bar{N}} \\ P_{2\bar{1}} & P_{2\bar{2}} & \cdots & P_{2\bar{N}} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N\bar{1}} & P_{N\bar{2}} & \cdots & P_{N\bar{N}} \end{bmatrix}$$

$$\{P_{\bar{\imath}j}\} = \begin{bmatrix} P_{\bar{1}1} & P_{\bar{1}2} & \cdots & P_{\bar{1}N} \\ P_{\bar{2}1} & P_{\bar{2}2} & \cdots & P_{\bar{2}N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{\bar{N}1} & P_{\bar{N}2} & \cdots & P_{\bar{N}N} \end{bmatrix}$$

$$\{P_{\bar{\imath}\bar{\jmath}}\} = \left[ \begin{array}{cccc} P_{\bar{1}\bar{1}} & P_{\bar{1}\bar{2}} & \cdots & P_{\bar{1}\bar{N}} \\ P_{\bar{2}\bar{1}} & P_{\bar{2}\bar{2}} & \cdots & P_{\bar{2}\bar{N}} \\ \vdots & \vdots & \ddots & \vdots \\ P_{\bar{N}\bar{1}} & P_{\bar{N}\bar{2}} & \cdots & P_{\bar{N}\bar{N}} \end{array} \right]$$

*Proof:* The given loops implement the product of a $2N \times 1$ matrix and a $1 \times 2N$ matrix, i.e., an outer product. Assume the input vector to be

$$\left( \begin{array}{ccccccccc} x_1 & x_2 & \cdots & x_N & | & x_{\bar{1}} & x_{\bar{2}} & \cdots & x_{\bar{N}} \end{array} \right)^T$$

Then the above loops perform

$$\left( \begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_N \\ \hline x_{\bar{1}} \\ x_{\bar{2}} \\ \vdots \\ x_{\bar{N}} \end{array} \right) \left( \begin{array}{ccccccccc} x_1 & x_2 & \cdots & x_N & | & x_{\bar{1}} & x_{\bar{2}} & \cdots & x_{\bar{N}} \end{array} \right)$$

to create a matrix of the form:

$$\mathbf{P} = \left[ \begin{array}{cc} \{P_{ij}\} & \{P_{i\bar{\jmath}}\} \\ \{P_{\bar{\imath}j}\} & \{P_{\bar{\imath}\bar{\jmath}}\} \end{array} \right].$$

■

Using the above definition for $\mathbf{P}$ matrix, we can obtain the following lemma.

*Lemma 3: If*

$$\mathbf{P} = \left[ \begin{array}{cc} \{P_{ij}\} & \{P_{i\bar{\jmath}}\} \\ \{P_{\bar{\imath}j}\} & \{P_{\bar{\imath}\bar{\jmath}}\} \end{array} \right]$$

*is defined in the same way as Lemma 2, we have the following:*

1) *The submatrices, $\{P_{i\bar{\jmath}}\}$, and, $\{P_{\bar{\imath}j}\}$, are symmetric.*
2) *The elements in the upper triangle of $\{P_{ij}\}$ are equal to those in the lower triangle of $\{P_{\bar{\imath}\bar{\jmath}}\}$. and that the elements in the lower triangle of $\{P_{ij}\}$ are equal to those in the upper triangle of $\{P_{\bar{\imath}\bar{\jmath}}\}$,* i.e. *that $\{P_{ij}\} = \{P_{\bar{\imath}\bar{\jmath}}\}^T$.*

*Proof:* From the Lemma 1 we know that $P_{ij} = P_{\bar{\jmath}\bar{\imath}}$. Therefore,

$$\{P_{ij}\} = \{P_{\bar{\imath}\bar{\jmath}}\}^T.$$

Now we prove that the submatrix, $\{P_{\bar{\imath}j}\}$, is symmetric. In equation $P_{ij} = P_{\bar{\jmath}\bar{\imath}}$, if we substitute $\bar{\imath}$ for $i$, we have $P_{\bar{\imath}j} = P_{\bar{\jmath}i}$. We will use this relation in the following proof. The submatrix, $\{P_{\bar{\imath}j}\}$, is located in the lower-left quadrant of $\mathbf{P}$. Imagine that we move this submatrix to the upper-left quadrant to overwrite the original upper-left one. This movement cannot change the symmetric property of the submatrix $\{P_{\bar{\imath}j}\}$ because we do not change the values of the elements of the submatrix. Recall that we arrange the set of input edges, $S$, so that the order of the first half of the $2N$ elements is the same as the input order (i.e., edge $i$'s) and the second $N$ of the $2N$ elements are edges with opposite direction to each element in the first half (i.e., edge $\bar{\imath}$'s). Therefore the index $\bar{\imath} = i + N$, and the index $\bar{\jmath} = j + N$. From $P_{\bar{\imath}j} = P_{\bar{\jmath}i}$, we have $P_{(i+N)j} = P_{(j+N)i}$. If we perform the movement just described, we have $P_{(i+N-N)j} = P_{(j+N-N)i}$, which is $P_{ij} = P_{ji}$. Therefore, the shifted submatrix is symmetric. It follows that the original submatrix is symmetric. Similarly, we can show that $\{P_{i\bar{\jmath}}\} = \{P_{i\bar{\jmath}}\}^T$. ■

*Theorem 1: Suppose we use the method described in Lemma 2 to construct the affinity matrix $\mathbf{P}$ and $\mathbf{x}$ is an eigenvector of $\mathbf{P}$ associated with eigenvalue, $\lambda$, where*

$$\mathbf{x} = \left( \begin{array}{ccccccc} x_1 & \cdots & x_N & | & x_{N+1} & \cdots & x_{2N} \end{array} \right)^T$$

*Then the eigenvector of $\mathbf{P}^T$ associated with the same eigenvalue $\lambda$ is*

$$\left( \begin{array}{ccccccccc} x_{N+1} & x_{N+2} & \cdots & x_{2N} & | & x_1 & x_2 & \cdots & x_N \end{array} \right)^T$$

*Proof:* Since we use the same method as Lemma 2 to construct the affinity matrix, it follows that

$$\mathbf{P} = \left[ \begin{array}{cc} \{P_{ij}\} & \{P_{i\bar{\jmath}}\} \\ \{P_{\bar{\imath}j}\} & \{P_{\bar{\imath}\bar{\jmath}}\} \end{array} \right]$$

where $\{P_{ij}\}$, $\{P_{i\bar{\jmath}}\}$, $\{P_{\bar{\imath}j}\}$ and $\{P_{\bar{\imath}\bar{\jmath}}\}$ are four $N \times N$ matrix whose elements are $P_{ij}$, $P_{i\bar{\jmath}}$, $P_{\bar{\imath}j}$ and $P_{\bar{\imath}\bar{\jmath}}$ respectively. Let $\mathbf{x}$ be an eigenvector of the matrix, $\mathbf{P}$,

$$\mathbf{x} = \left( \begin{array}{c} \mathbf{x}_1 \\ \mathbf{x}_2 \end{array} \right)$$

where $\mathbf{x_1}$ consists of the first $N$ elements of the vector $\mathbf{x}$ and $\mathbf{x_2}$ consists of the second $N$ elements of the vector $\mathbf{x}$.
From $\mathbf{Px} = \lambda\mathbf{x}$, we have

$$\left[ \begin{array}{cc} \{P_{ij}\} & \{P_{i\bar{\jmath}}\} \\ \{P_{\bar{\imath}j}\} & \{P_{\bar{\imath}\bar{\jmath}}\} \end{array} \right] \left( \begin{array}{c} \mathbf{x}_1 \\ \mathbf{x}_2 \end{array} \right) = \lambda \left( \begin{array}{c} \mathbf{x}_1 \\ \mathbf{x}_2 \end{array} \right),$$

which is equivalent to

$$\left[ \begin{array}{cc} \{P_{\bar{\imath}\bar{\jmath}}\} & \{P_{\bar{\imath}j}\} \\ \{P_{i\bar{\jmath}}\} & \{P_{ij}\} \end{array} \right] \left( \begin{array}{c} \mathbf{x}_2 \\ \mathbf{x}_1 \end{array} \right) = \lambda \left( \begin{array}{c} \mathbf{x}_2 \\ \mathbf{x}_1 \end{array} \right).$$

From Lemma 3, we obtain

$$\left[ \begin{array}{cc} \{P_{\bar{\imath}\bar{\jmath}}\} & \{P_{\bar{\imath}j}\} \\ \{P_{i\bar{\jmath}}\} & \{P_{ij}\} \end{array} \right] = \left[ \begin{array}{cc} \{P_{ij}\}^T & \{P_{\bar{\imath}j}\}^T \\ \{P_{i\bar{\jmath}}\}^T & \{P_{\bar{\imath}\bar{\jmath}}\}^T \end{array} \right].$$

Therefore,

$$\left[ \begin{array}{cc} \{P_{ij}\}^T & \{P_{\bar{\imath}j}\}^T \\ \{P_{i\bar{\jmath}}\}^T & \{P_{\bar{\imath}\bar{\jmath}}\}^T \end{array} \right] \left( \begin{array}{c} \mathbf{x}_2 \\ \mathbf{x}_1 \end{array} \right) = \lambda \left( \begin{array}{c} \mathbf{x}_2 \\ \mathbf{x}_1 \end{array} \right).$$

Finally, since

$$\mathbf{P}^T = \left[ \begin{array}{cc} \{P_{ij}\}^T & \{P_{\bar{\imath}j}\}^T \\ \{P_{i\bar{\jmath}}\}^T & \{P_{\bar{\imath}\bar{\jmath}}\}^T \end{array} \right]$$

we finish our proof. ■

Note that it is necessary to use $\bar{\imath}$ notation to represent some elements of the eigenvectors in order to define link saliency in the next section. We use $\mathbf{x}$ to represent the eigenvector of the matrix, $\mathbf{P}$, and $\bar{\mathbf{x}}$ to represent the eigenvector of the matrix, $\mathbf{P}^T$. In addition, we use $i$ to represent the index which is between 1 and $N$, and $\bar{\imath}$ to represent the index which is between $N$ and $2N$. For example, $\bar{x}_{\bar{\imath}}$ is the $(i+N)$-th element of the eigenvector of the $\mathbf{P}^T$ matrix. With this notation, we obtain $\bar{s}_i = s_{\bar{i}}$ and $\bar{s}_{\bar{i}} = s_i$ directly from the above theorem.

Now we can simplify the edge saliency definition. Recall that

$$c_i = \frac{x_i \bar{x}_i}{\sum_j x_j \bar{x}_j}$$

which involves both the right and left eigenvectors. With the relation between $\mathbf{x}$ and $\bar{\mathbf{x}}$, we can rewrite the edge saliency definition as

$$c_i = \frac{x_i x_{(i \pm N) \bmod N}}{\sum_j x_j x_{(j \pm N) \bmod N}}$$

which is only dependent on the right eigenvector with largest positive real eigenvalue of affinity matrix, $\mathbf{P}$. Without solving the left eigenvector, we can save much running time.

Actually, we can reduce computation further using the following theorem.

*Theorem 2: For the* $\mathbf{P}$ *matrix, there are only* $2N^2 - 2N + 2$ *distinct elements.*

*Proof:* This can be demonstrated quite easily. From Lemma 3, we know that submatrices $\{P_{i\bar{j}}\}$ and $\{P_{\bar{i}j}\}$ are symmetric. We also know that the elements in the upper triangle of $\{P_{ij}\}$ are equal to those in the lower triangle of $\{P_{\bar{i}\bar{j}}\}$, and that the elements in the lower triangle of $\{P_{ij}\}$ are equal to those in the upper triangle of $\{P_{\bar{i}\bar{j}}\}$. Therefore, if we do not consider the diagonal elements of the four matrices, then the total number of distinct elements is $2N^2 - 2N$. In addition, we also can see that the diagonal elements of matrix $\mathbf{P}$ are constant and so are the diagonal elements of the submatrices $\{P_{i\bar{j}}\}$ and $\{P_{\bar{i}j}\}$. These results are the consequences of Lemma 1. Finally, we have $2N^2 - 2N + 2$ distinct elements. ■

It should be note that the idea of Theorem 1 was first introduced in [14]. Here we give the formal proof as complement. Theorem 2 is new and has not been presented before.

### D. Link Saliency

Suppose we begin at edge $i$, immediately visit edge $j$, and then eventually return to $i$. We have the following mathematical expression:

$$C_{ij} = \lim_{n \to \infty} \frac{P^{n-1}(j|i)P(i|j)}{\sum_k P^{(n)}(k|k)}.$$

Williams and Thornber[14] solved this expression

$$C_{ij} = \frac{\bar{x}_i P_{ij} x_j}{\lambda \bar{\mathbf{x}}^T \mathbf{x}}.$$

If we use the approach mentioned in the previous sections and we normalize the eigenvectors so that $\sum_i x_i \bar{x}_i = 1$, we can simplify the above expression as follows:

$$C_{ij} = \frac{x_{(i+N) \bmod N} P_{ij} x_j}{\lambda}.$$

which is only dependent on the right eigenvector.

As with the $\mathbf{P}$ matrix, the $\mathbf{C}$ matrix has many interesting properties.

*Theorem 3: The* $\mathbf{C}$ *matrix created using the above equation has the following properties:*

1) *Conservation property: The number of closed contours entering edge i equals the number of closed contours leaving edge i.*
2) *Reversal symmetry property:* $C_{ij} = C_{\bar{j}\bar{i}}$
3) *In the above equation, we can substitute* $i$ *for* $\bar{i}$ *everywhere, and* $\bar{i}$ *for* $i$ *everywhere.*

*Proof:* For convenience, we suppose $\sum_i x_i \bar{x}_i = 1$. This assumption involves no loss of generality because any eigenvector can be normalized so that it has unit length. Property *1* has been shown by Williams and Thornber[14]. We prove Properties 2 and 3. Recall that $P_{ij} = P_{\bar{j}\bar{i}}$, $\bar{x}_i = x_{\bar{i}}$ and $\bar{x}_{\bar{i}} = x_i$. Thus, it is easy to verify that

$$C_{ij} = \frac{\bar{x}_i P_{ij} x_j}{\lambda} = \frac{x_{\bar{i}} P_{\bar{j}\bar{i}} \bar{x}_{\bar{j}}}{\lambda} = \frac{\bar{x}_{\bar{j}} P_{\bar{j}\bar{i}} x_{\bar{i}}}{\lambda} = C_{\bar{j}\bar{i}}.$$

In $P_{ij} = P_{\bar{j}\bar{i}}$, we can substitute $i$ for $\bar{i}$ everywhere, and $\bar{i}$ for $i$ everywhere. Thus, we can obtain $P_{i\bar{j}} = P_{j\bar{i}}$ and $P_{\bar{i}j} = P_{\bar{j}i}$ directly from $P_{ij} = P_{\bar{j}\bar{i}}$. Therefore,

$$C_{i\bar{j}} = \frac{\bar{x}_i P_{i\bar{j}} x_{\bar{j}}}{\lambda} = \frac{x_{\bar{i}} P_{j\bar{i}} \bar{x}_j}{\lambda} = \frac{\bar{x}_j P_{j\bar{i}} x_{\bar{i}}}{\lambda} = C_{j\bar{i}}.$$

Similarly, we can show that $C_{\bar{i}j} = C_{\bar{j}i}$, and thus Property 3 is proven. ■

## III. Segmentation Algorithm

Mahamud *et al.*[7] use an approach based on a global saliency measure, called *link saliency*, which is related to the relative likelihood that closed contours pass through edges $j$ and $i$ successively. They first compute the affinity matrix, $P$, and the eigenvector associated with its maximum eigenvalue, and then compute the *link saliency* matrix, $C$. Recall that the *link saliency* is the relative number of closed contours which visit edge $i$ and $j$ in succession. It provides a measure of whether two edges are salient, i.e., the likelihood that a contour passes through the pair of edges.Thereore, the *link saliency* matrix is used to create a directed graph. The segmentation problem is reduced to the problem of finding strongly-connected components of this graph. They can be found using generic graph search techniques such as depth-first-search or modified depth-first-search.

We can find all the strongly-connected components at once. Unfortunately, not all of the components found in this way provide reliable segmentations. The reason is that the dominant contours tend to suppress the saliencies of all of the other contours, and therefore its saliencies suppress the saliencies of less dominant contours. The other way is to extract out the contours one by one. We first extract out the most dominant contour which passes through the most salient edge. The strongly-connected component containing this most dominant contour is the set of edges reachable from the most salient edge. This set of edges can be easily found using depth-first-search of the directed graph. After extracting out the dominant component, we suppress it to find the next dominant component. To suppress the current component, we deflate

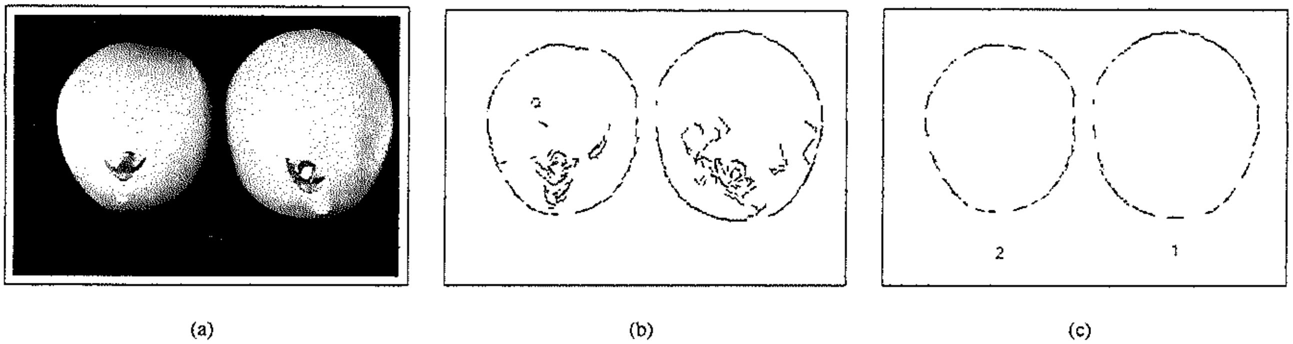


Fig. 1. *Double Pears.* (a) Original image. (b) Edge input obtained by Canny detector. (c) Segmented contours numbered in the order in which they are extracted.

the affinities of all links among the edges in the strongly-connected component. For example, if edges, *i* and *j* are edges in the component, we deflate the link $i \to j$ by setting $P_{ij} = 0$ and $P_{\bar{j}\bar{i}} = 0$. We iterate the process of depth-first search followed by deflation until all components are isolated.

Due to noise, strongly-connected components which include edges of the current most salient object may also contain edges which belong to less salient objects or to the background. We can eliminate these undesirable edges using an important property of strongly-connected components, namely that, *the transpose a directed graph $G = (V, E)$ has exactly the same strong connected components as those of G.* Using the reversal-symmetry property, we can implement this operation as follows[7]: (1) Find $A$, the set of edges reachable from the most salient edge, $i$; (2) Find $\bar{A}$, the set of edges reachable from the edge, $\bar{\imath}$; and (3) Find the intersection of the two sets, $A \cap \bar{A}$.

## IV. Experimental Results

To verify the properties of the affinity matrix and algorithm, we conducted experiments with two real images. In the first segmentation experiment,we did not consider the properties of the affinity matrix when we built it and solved its eigenvalue problem. The second segmentation experiment is the same as the first one except that at this time we considered the properties of the affinity matrix when building the matrix, and finding its eigenvalue and eigenvector. The number of edges returned by the Canny detector was found to be prohibitively large. To reduce the running time, we subsampled the input edges with no sacrifice in comparison. After the input edges were sampled, the two affinity matrix sizes are 898 × 898 and 3044 × 3044. The three parameters which determine the distribution shapes are $\gamma = 0.15$, $T = 0.004$, and $\tau = 5.0$. We repeated each experiment 10 times and found the average running times. In the first experiment, it took about *4.37 seconds* to segment each pear, and about *34.20 seconds* to segment each coin while in the second experiment it only took *2.28 seconds*, and *17.24 seconds* respectively. The average running times were reduced to 47.9% for *Pears Segmentation*, and to 49.6% for *Coins Segmentation*. Because the matrix size for *coins segmentation* is larger than that for *pears segmentation*, it took more time to segment each coin. The original image, edge input obtained by Canny detector, and segmented contours output numbered in the order which they are extracted for these experiments are shown in Figure 1 and Figure 2.

## V. Conclusions

The purpose of segmentation is to partition a set of edges into subsets so that the edges in different subsets bound different objects in the scene. Thornder and Williams[14]'s analytic expression which characterizes the probability distribution of boundary-completion shapes provides the basic foundation for solving the segmentation problem. In this paper, we analyzed and proved the properties of the affinity matrix of Williams and Thornber. The experimental results show that using these properties we can reduce up to 50% running time. This was significant for improving computational efficiency. It is also significant in robot visual servo, which needs quick response. In the future, we will try to improve efficiency further and apply this algorithm into visual robot control system to pick multiple objects in the scene.

## Acknowledgments

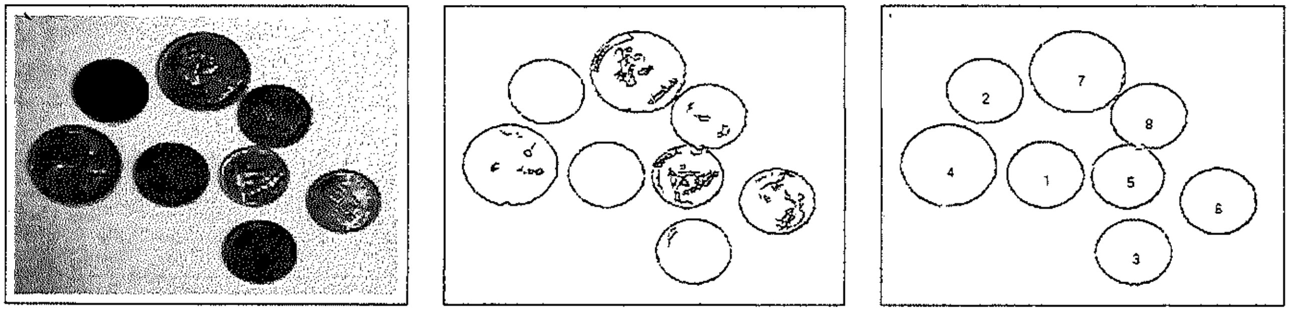
The authors would like to thank Lance Williams, Karvel Thornber and Shyjan Mahamud for their contributions and advices. Lance williams and Karvel Thornber first present and use the some properties of the affinity matrix to reduce the amount of computation. They also give the approach to calculate the *edge saliency*, the *link saliency*, and the *affinity matrix*. Shyjan Mahamud presents the algorithm to segment the multiple salient closed contours from real images. This research was partially supported by Los Alamos National Laboratory.

## References

[1] Chatelin, F., *Eigenvalues of Matrices*, Chichester, New York, 1993.

[2] Elder, J.H. and Zucker, S.W., A Measure of Closure, *Vision Research*, vol. 34, no. 24, pp. 3361-3370, 1994.

[3] Elder, J.H. and Zucker, S.W., Computing Contour Closure, *ECCV'96*, Cambridge, UK. Vol. 1, p14-18, 1996.



(a) (b) (c)

Fig. 2. *Coins*. (a) Original image. (b) Edge input obtained by Canny detector. (c) Segmented contours numbered in the order in which they are extracted.

[4] J. Freixenet, X. Muoz, D. Raba, J. Mart, and X. Cuf, Yet Another Survey on Image Segmentation: Region and Boundary Information Integration, *Proc. ECCV02*, LNCS 2353, 408 ff., Springer, 2002.

[5] M. Gong and Y.H. Yang, Genetic-based multiresolution color image segmentation, *Vision Interface 2001*, June 7-9, 2001, Ottawa, Ontario, pp. 141-148.

[6] T. Leung and J. Malik, Contour continuity in region based image segmentation, *Fifth European Conference on Computer Vision*, June 1998.

[7] S.Mahamud, L.Williams, K.Thornber, Kanglin Xu, Segmentation of Multiple Salient Closed Contours from Real Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 4, April 2003.

[8] A.M. Martinez, P. Mittrapiyanuruk and A.C. Kak, On Combining Graph-Partitioning with Non-Parametric Clustering for Image Segmentation, *Computer Vision and Image Understanding*, Vol. 95, No. 1, pp. 72-85, 2004.

[9] Mumford, D., Elastica and Computer Vision, *Algebraic Geometry and Its Application*, Chandrajit Bajaj (ed.), Springer-Verlag, New York, 1994.

[10] Parker, J.R., *Algorithms for Image Processing and Computer Vision*,Wiley Computer Publishing, John Wiley & Sons, Inc., 1997.

[11] Perona, P. and Freeman, W., A Factorization Approach to Grouping, *Proc. 5th European Conference of Computer Vision (ECCV98)*, Freiburg, Germany, Pages 655-670, 1998.

[12] X. Ren and J. Malik, Learning a Classification Model for Segmentation, *CVPR*, 2003.

[13] Shi, J. and Malik, J., Normalized Cuts and Image Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI)*, 2000.

[14] Williams, L.R. and Thornber, K.K, A Comparison of Measures for Detecting Natural Shapes in Cluttered Backgrounds, *Inter. J. Computer Vision* 34(2/3), 81-96, 2000.

[15] Stella Yu, Ralph Gross and Jianbo Shi, Concurrent Object Segmentation and Recognition with Graph Partitioning, *Neural Information Processing Systems*, December 2002.

# ICAR 2005

Home

Technical Program
UPDATED

Plenary Speakers

Workshops and Tutorials

Conference Registration
Early Fee deadline: June 1

Social Events

Attractions

Location & Lodging

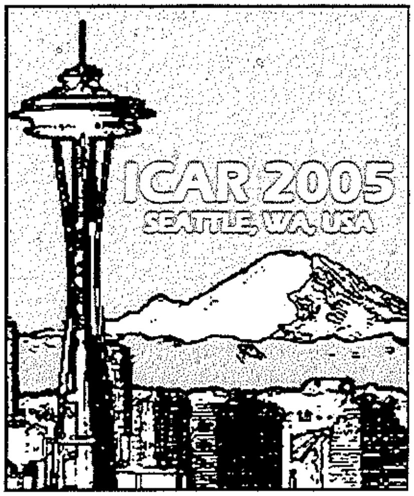
Poster

Call for Papers

Organizing Committee

Program Committee

ICAR2005 is technically co-sponsored by IEEE Robotics and Automation Society



*"A Multitude of Robot Species: Applications, Ubiquity, and Biological Inspiration"*

Welcome to the 12th International Conference on Advanced Robotics. ICAR 2005 will be held in Seattle, Washington, USA, Monday July 18th - Wednesday July 20th, 2005. Workshops and tutorials are to be held Sunday, July 17th. Plenaries will be given by Prof. Sebastian Thrun, Stanford University, Prof. Richard Satava M.D., Univ of Washington, and Dr. Paul S. Schenker, Jet Propulsion Laboratory. The conference will take place in the Seattle Sheraton Hotel and Towers located in beautiful downtown Seattle. The conference banquet will take place on Tuesday evening at the Kiana Lodge (will include cruise). A special discount on an Alaskan Cruise post-conference is available to all conference attendees.

Blake Hannaford
ICAR'05 General Chair

Technical Program
Plenary Speakers
Workshops & Tutorials
Registration

Kiana Lodge Banquet
Alaskan Cruise

### Important dates:

| | |
|---|---|
| Accepted Papers due to Publisher | *10-May-2005* |
| Discounted Early Registration | **01-June-2005** |

**Supported by:** Microsoft and **BOEING**