Image Segmentation using Commute times

Huaijun Qiu and Edwin R. Hancock
Department of Computer Science, University of York,
York, YO10 5DD, UK

Abstract

This paper exploits the properties of the commute time to develop a graph-spectral method for image segmentation. Our starting point is the lazy random walk on the graph, which is determined by the heat-kernel of the graph and can be computed from the spectrum of the graph Laplacian. We characterise the random walk using the commute time between nodes, and show how this quantity may be computed from the Laplacian spectrum using the discrete Green's function. We explore the application of the commute time for image segmentation using the eigenvector corresponding to the smallest eigenvalue of the commute time matrix.

1 Introduction

Spectral graph theory [2] is concerned with characterising the structural properties of graphs using information conveyed by the eigenvalues and eigenvectors of the Laplacian matrix (the degree matrix minus the adjacency matrix). One of the most important tasks that arises in the analysis of graphs is that of how information flows with time across the edges connecting nodes. This process can be characterised using the heat equation [5]. The solution of the heat equation, or heat kernel, can be found by exponentiating the Laplacian eigensystem over time. The heat kernel contains a considerable amount of information concerning the distribution of paths on the graph. For instance, it can be used to compute the lazy random walk on the nodes of the graph. It may also be used to determine commute times under the random walk between pairs of nodes. An alternative, but closely related, characterisation of the graph is the discrete Green's function which captures the distribution of sources in the heat flow process. Not surprisingly, there is a direct link between commute times and the Green's function [3].

Random walks [14] have found widespread use in information retrieval and structural pattern analysis. For instance, the random walk is the basis of the Page-Rank algorithm which is used by the Googlebot search engine [1]. In computer vision random walks have been used for image segmentation [7] and clustering [10]. More recently both Gori, Maggini and Sarti [4], and, Robles-Kelly and Hancock [9] have used random walks to sort the nodes of graphs for the purposes of graph-matching. However, most of these methods use a simple approximate characterisation of the random walk based either on the leading eigenvector of the transition probability matrix, or equivalently the Fiedler vector of the Laplacian matrix [6]. However, a single eigenvector can not be used to determine more detailed information concerning the random walk such as the distribution of commute times. The aim in this paper is to draw on more detailed information contained within the Laplacian spectrum, and to use the commute time as means of grouping.

There are two quantities that are commonly used to define the utility in graph-theoretic methods for grouping and clustering. The first of these is the association, which is a measure of total edge linkage within a cluster and is useful in defining clump structure. The second is the cut, which is a measure of linkage between different clusters and can be used to split extraneous nodes from a cluster. Several methods use eigenvectors to extract clusters using the utility measure. Some of the earliest work was done by Scott and Longuet-Higgins [12] who developed a method for refining the block-structure of the affinity matrix by relocating its eigenvectors. At the level of image segmentation, several authors have used algorithms based on the eigenmodes of an affinity matrix to iteratively segment image data. For instance, Sarkar and Boyer [11] have a method which uses the leading eigenvector of the affinity matrix, and this locates clusters that maximise the average association. This method is applied to locating line-segment groupings. The method of Shi and Malik [13], on the other hand, uses the normalized cut which balances the cut and the association. Clusters are located by performing a recursive bisection using the eigenvector associated with the second smallest eigenvalue of the Laplacian (the degree matrix minus the adjacency matrix), i.e. the Fiedler vector. Recently Pavan and Pelillo [8] have shown how the concept of a dominant set can lead to better defined clusters, and can give results that are superior to those delivered by the Shi and Malik algorithm for image segmentation. The dominant set provides a more subtle definition of cluster membership that draws on the mutual affinity of nodes. The method does not rely simply on the affinity between pairs of nodes alone. Here we argue that commute time can also capture the affinity properties of nodes in a way that extends beyond the use of pairwise weights.

Graph theoretic methods aim to locate clusters of nodes that minimize the cut or disassociation, while maximizing the association. The commute time has properties that can lead to clusters of nodes that increase both the dissociation and the association. A pair of nodes in the graph will have a small commute time value if one of three conditions is satisfied. The first of these is that they are close together, i.e. the length of the path between them is small. The second case is if the sum of the weights on the edges connecting the nodes is small. Finally, the commute time is small if the pair of nodes are connected by many paths. Hence, the commute time can lead to a finer measure of cluster cohesion than the simple use of edge-weight which underpins algorithms such as the normalized cut [13]. In this respect it is more akin with the method of Pavan and Pelillo [8].

2 Heat Kernel, Lazy Random Walks and Green's Function

2.1 Heat kernel

Let the weighted graph Γ be the quadruple (V, E, Ω, ω), where V is the set of nodes, E is the set of arcs, $\Omega = \{W_u, \forall u \in V\}$ is a set of weights associated with the nodes and $\omega = \{w_{u,v}, \forall (u,v) \in E\}$ is a set of weights associated with the edges. Further let $T = diag(d_v; v \in V(\Gamma))$ be the diagonal weighted degree matrix with $T_u = \sum_{v=1}^{n} w_{u,v}$. The unnormalized weighted Laplacian matrix is given by $L = T - A$ and the normalized weighted Laplacian matrix is defined to be $\mathscr{L} = T^{-1/2} L T^{-1/2}$, and has elements

$$\mathscr{L}_{uv}(\Gamma) = \begin{cases} 1 & \text{if } u = v \\ -\frac{w_{u,v}}{\sqrt{d_u d_v}} & \text{if } u \neq v \text{ and } (u,v) \in E \\ 0 & \text{otherwise} \end{cases}$$

The spectral decomposition of the normalized Laplacian is $\mathscr{L} = \Phi \Lambda \Phi^T$, where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_{|V|})$ is the diagonal matrix with the ordered eigenvalues as elements satisfying: $0 = \lambda_1 \leq \lambda_2 \ldots \leq \lambda_{|V|}$ and $\Phi = (\phi_1|\phi_2|....|\phi_{|V|})$ is the matrix with the ordered eigenvectors as columns.

In the paper we are interested in the heat equation associated with the graph Laplacian, i.e. $\frac{\partial \mathscr{H}_t}{\partial t} = -\mathscr{L}\mathscr{H}_t$ where $\mathscr{H}_t$ is the heat kernel and t is time. The solution of the heat-equation is found by exponentiating the Laplacian eigenspectrum i.e.$\mathscr{H}_t = \exp[-t\mathscr{L}] = \Phi\exp[-t\Lambda]\Phi^T$. The heat kernel is a $|V| \times |V|$ matrix, and for the nodes u and v of the graph Γ the element of the matrix is $\mathscr{H}_t(u,v) = \sum_{i=1}^{|V|} \exp[-\lambda_i t]\phi_i(u)\phi_i(v)$.

2.2 Lazy random walk

Let us consider the matrix $\mathscr{P} = T^{1/2}PT^{-1/2} = I - \mathscr{L}$, where I is the identity matrix, then we can re-express the heat kernel by performing a McLaurin expansion as(see [3] page 5),

$$\mathscr{H}_t = e^{-t(I-\mathscr{P})} = e^{-t}\sum_{r=1}^{\infty}\mathscr{P}^r\frac{t^r}{r!}$$

Using the spectral decomposition of the normalized Laplacian, we have $\mathscr{P}^r = (I-\mathscr{L})^r = \Phi(I-\Lambda)^r\Phi^T$ and as a result

$$\mathscr{P}^r(u,v) = \sum_{i=1}^{|V|}(1-\lambda_i)^r\phi_i(u)\phi_i(v) = \sum_{\pi_r}\prod_i\frac{w(u_i,u_{i+1})}{\sqrt{d_{u_i}d_{u_{i+1}}}}$$

Lemma 2.1 *The normalized probability matrix $\mathscr{P}^r(u,v)$ is the sum of the probabilities of all the random walks π of length r connecting node u and v.*

Theorem 2.2 *The heat kernel is the continuous time limit of the lazy random walk.*

Proof *Consider a lazy random walk $R = (1-\alpha)I + \frac{W}{T}\alpha$ which migrates between different nodes with probability α and remains static at a node with probability $1-\alpha$, where W is the weighted adjacency matrix and T is the degree matrix.*

Let $\alpha = \alpha_0\Delta t$ where $\Delta t = \frac{1}{N}$. Consider the distribution $R(V_N|V_0)$, which is the probability of the random walk joining node 0 and N, in the limit $\Delta t \to 0$

$$\lim_{N\to\infty} R^N = \lim_{N\to\infty}\left(I + (\frac{W}{T} - I)\alpha_0\frac{1}{N}\right)^N = e^{(\frac{W}{T}-I)\alpha_0} \tag{1}$$

while

$$\frac{W}{T} - I = T^{-1}A - I = T^{-1}(T-L) - I = -T^{-1}L \tag{2}$$

Now consider the discrete Laplace operator Δ with the following properties: $\mathscr{L} = T^{1/2}\Delta T^{-1/2} = T^{-1/2}LT^{-1/2}$, which implies $\Delta = LT^{-1}$. As a result, we get $\lim_{N\to\infty} R^N = e^{-\Delta\alpha_0}$ which is just the expression for the heat kernel. ∎

2.3 Green's function

Now consider the discrete Laplace operator $\Delta = T^{-1/2}\mathscr{L}T^{1/2}$. The Green's function is the left inverse operator of the Laplace operator Δ, defined by $G\Delta(u,v) = I(u,v) - \frac{d_v}{vol}$, where $vol = \sum_{v\in V(\Gamma)} d_v$ is the volume of the graph. A physical interpretation of the Green's function is the temperature at a node in the graph due to a unit heat source applied to the external node. It is related with the heat kernel $\mathscr{H}_t$ in the following manner

$$G(u,v) = \int_0^{\infty} d_u^{1/2} \left(\mathscr{H}_t(u,v) - \phi_1(u)\phi_1(v)\right) d_v^{-1/2} dt \tag{3}$$

Here ϕ_1 is the eigenvector associated with eigenvalue 0 and its k-th entry is $\sqrt{d_k/vol}$. Furthermore, the normalized Green's function $\mathscr{G} = T^{-1/2}GT^{1/2}$ is defined as (see [3] page 6(10)),

$$\mathscr{G}(u,v) = \sum_{i=2}^{|V|} \frac{1}{\lambda_i} \phi_i(u)\phi_i(v) \tag{4}$$

where λ and ϕ are the eigenvalue and eigenvectors of the normalized Laplacian $\mathscr{L}$.

The normalized Green's function is hence the generalized inverse of the normalized Laplacian $\mathscr{L}$. Moreover, it is straightforward to show that $\mathscr{G}\mathscr{L} = \mathscr{L}\mathscr{G} = I - \phi_1\phi_1^*$, and as a result $(\mathscr{L}\mathscr{G})_{uv} = \delta_{uv} - \frac{\sqrt{d_u d_v}}{vol}$. From equation 4, the eigenvalues of $\mathscr{L}$ and $\mathscr{G}$ have the same sign and $\mathscr{L}$ is positive semidefinite, and so $\mathscr{G}$ is also positive semidefinite. Since $\mathscr{G}$ is also symmetric(see [3] page 4), it follows that $\mathscr{G}$ is a kernel.

3 Commute Time

We note that the *hitting time* $Q(u,v)$ of a random walk on a graph is defined as the expected number of steps before node v is visited, commencing from node u. The *commute time* $CT(u,v)$, on the other hand, is the expected time for the random walk to travel from node u to reach node v and then return. As a result $CT(u,v) = Q(u,v) + Q(v,u)$. The hitting time $Q(u,v)$ is given by [3]

$$Q(u,v) = \frac{vol}{d_v} G(v,v) - \frac{vol}{d_u} G(u,v)$$

where G is the Green's function given in equation 3. So, the commute time is given by

$$CT_{uv} = Q_{uv} + Q_{vu} = \frac{vol}{d_u} G_{uu} + \frac{vol}{d_v} G_{vv} - \frac{vol}{d_u} G_{uv} - \frac{vol}{d_v} G_{vu} \tag{5}$$

As a consequence of (5) the commute time is a metric on the graph. The reason for this is that if we take the elements of G as inner products defined in a Euclidean space, CT will become the norm satisfying: $\left\| x_i - x_j \right\|^2 = < x_i - x_j, x_i - x_j > = < x_i, x_i > + < x_j, x_j > - < x_i, x_j > - < x_j, x_i >$.

Substituting the spectral expression for the Green's function into the definition of the commute time, it is straightforward to show that

$$CT(u,v) = vol \sum_{i=2}^{|V|} \frac{1}{\lambda_i} \left(\frac{\phi_i(u)}{\sqrt{d_u}} - \frac{\phi_i(v)}{\sqrt{d_v}} \right)^2 \tag{6}$$

For a regular graph with $d_u = d_v = d$, and the commute time satisfies:

$$CT(u,v) = \frac{vol}{d} \sum_{i=2}^{|V|} \frac{1}{\lambda_i} (\phi_i(u) - \phi_i(v))^2 \tag{7}$$

This expression is important, since in the data clustering and image segmentation literature it is usual to work with an affinity matrix, and the underlying graph is therefore regular for the clustering problem and almost regular for the segmentation problem (boundary pixels have smaller degrees). As a result, the commute time can be taken as a generalisation of the normalized cut since from Equation 7, for a pair of node u and v the commute time depends on the difference of the components of the successive eigenvectors of $\mathscr{L}$. Of the eigenvectors, the Fiedler vector is the most significant since its corresponding eigenvalue λ_2 is the smallest.

4 Commute Times for Grouping

The idea of our segmentation algorithm is to use the spectrum of the commute time matrix for the purposes of grouping. We do this by using the eigenvector corresponding to the smallest eigenvalue to bipartition the graphs recursively.

Our commute time algorithm consists of the following steps:

1. Given an image, or a point set, set up a weighted graph $\Gamma = (V, E)$ where each pixel, or point, is taken as a node and each pair of nodes is connected by an edge. The weight on the edge is assigned according to the similarity between the two node as follows

   a) for a point-set, the weight between node i and j is set to be $w(i, j) = exp(-d(i, j)/\delta_x)$, where $d(i, j)$ is the Euclidean distance between two points and δ_x controls the scale of the spatial proximity of the points.

   b) for an image, the weight is:

$$w(i,j) = exp\left(\frac{-\left\|\mathbf{F}_i - \mathbf{F}_j\right\|_2}{\delta_I}\right) * \begin{cases} exp\left(\frac{-\left\|\mathbf{X}_i - \mathbf{X}_j\right\|_2}{\delta_X}\right) & \text{if } \left\|\mathbf{X}_i - \mathbf{X}_j\right\|_2 < r \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

   where $\mathbf{F}_i$ is the intensity value at pixel i for a brightness image or the RGB value for a color image.

2. From the weight matrix W we compute the Laplacian $L = T - W$.

3. Then we compute the normalized Green's function using Equation 4 and the eigenspectrum of the normalized Laplacian $\mathscr{L}$.

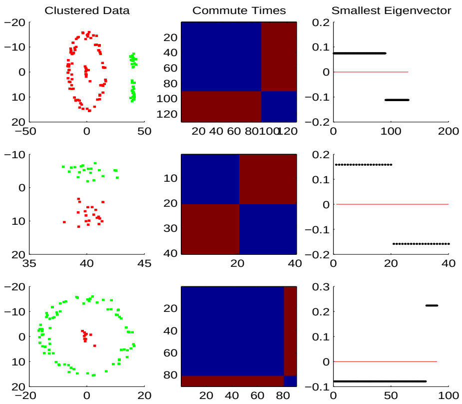
4. From Equation 5, we compute the commute time matrix CT whose elements are the commute times between each pair of nodes in the graph Γ.

5. Use the eigenvector corresponding to the smallest eigenvalue of the commute time matrix to bipartition the weighted graph.

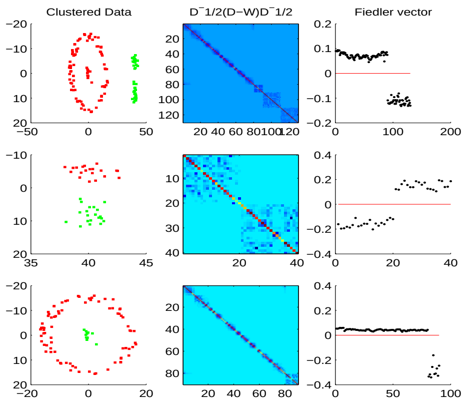
6. Decide if the current partition should be sub-divided, and recursively repartition the component parts if necessary.

5 Experiments

In this section we experiment with our new spectral clustering method. We commence with examples on synthetic images aimed at evaluating the noise sensitivity of the method.



(a) Data clustering by commute time.



(b) Data clustering by normalized cut.

Figure 1: Clustering examples.

We then provide examples on real world images and compare the performance of our method with that of Shi and Malik.

Point-set clustering examples: In Figure 1(a) and 1(b) we compare the results for point-set clustering using commute-times and the normalized cut. Here we set $\delta = 1.5$. The sub-figures in both figures are organised as follows. The left-hand column shows the point-sets, the middle column the affinity matrices and right-most column the components of the smallest eigenvector. The first row shows the first bipartition and the successive two rows show the bipartition based on the first partitions. From the figures it is clear that both methods succeeded in grouping the data. However, the commute time method outperforms the normalized cut since its affinity matrix is more block like and the distribution of the smallest eigenvector components is more stable, and its jumps corresponding to the different clusters in the data are larger.

Image segmentation: We have compared our new method with that of Shi and Malik [13] on synthetic images subject to additive Gaussian noise. On the left-hand side of Figure 2, we show the results of using these two methods for segmenting a synthetic image composed of 3 rectangular regions with additive Gaussian noise increasing from 0.04 to 0.20 with width 0.04. On the right hand side of Figure 2 we show the fraction of pixels correctly assigned as a function of the noise standard derivation. At the highest noise levels our method outperforms the Shi and Malik method by about 10%.

In Figure 3, we show some examples of our segmentation results and compare them with those obtained using the normalized cut. The aim here is to investigate the effect of adding and deleting link-weights at random. The first column shows the original image, the second column the original affinity matrix and the third colum the affinity matrix after link noise has been added. The first three rows show the effect of random link deletion, and the second three rows the result of link addition. The fourth and fifth columns show the results obtained using the normalized cut and the commute time. For these images, Figure 4 shows the fraction of correctly assigned pixels as a function of the fraction of links added or deleted. In the figure the red curve shows the effect of link addition on the commute time method, the green curve the effect of link addition on the normalized cut, the blue curve the effect of link deletion on the commute time method and, finally, the

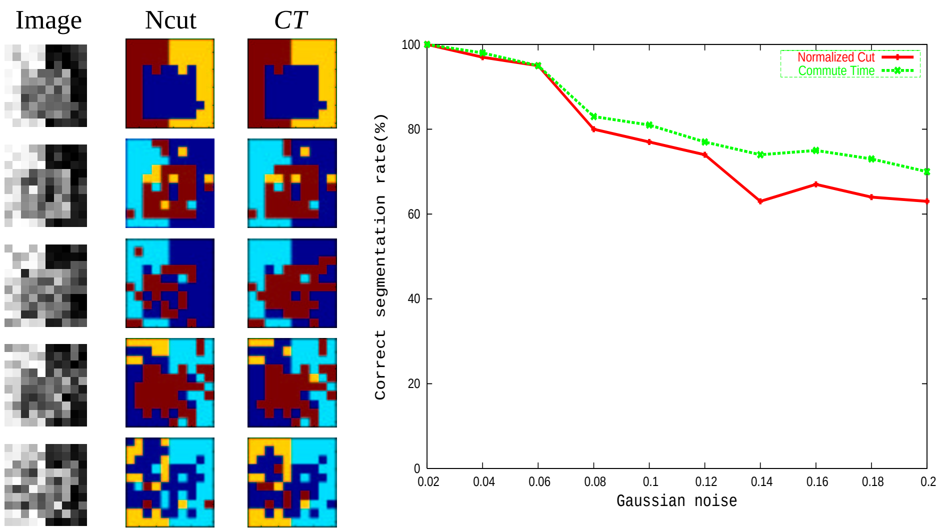


Figure 2: Method comparison for synthetic image with increasing Gaussian noise.

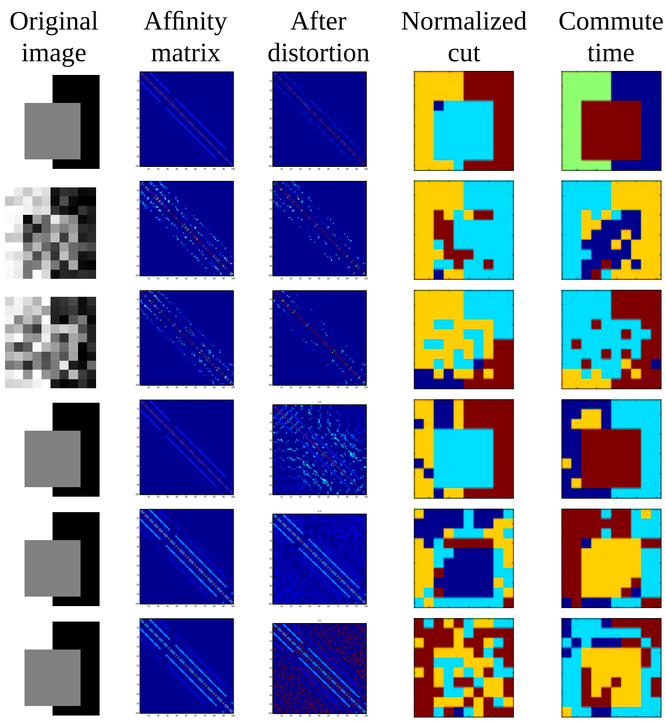


Figure 3: Examples of segmentation results with different link-weight distortion.

pink curve the effect of link deletion on the normalized cut. The main features to note from the plot are as follows. First, the commute time method is more robust to both link

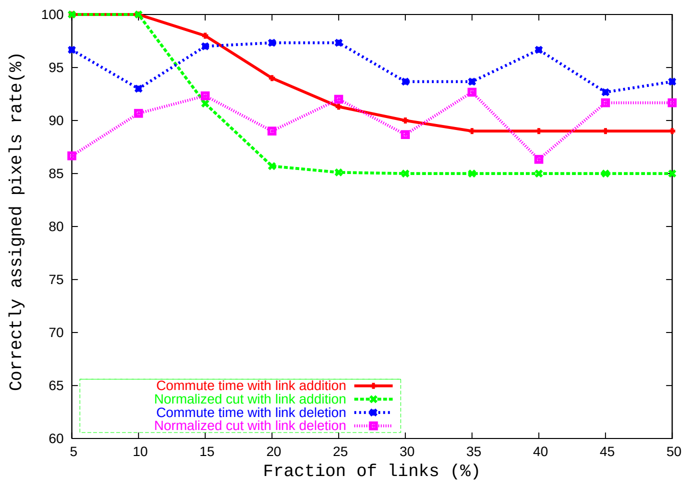


Figure 4: Method comparison for synthetic images with different link-weight distortion.

deletion and insertion than the normalized cut. The second feature is that link deletion has a less marked effect on the performance than link insertion. Thirdly, spurious link insertion has a smaller effect on the commute time than the normalized cut.

In Figure 5, we show eight real world images (from the Berkeley image database) with the corresponding segmentation results. The images are scaled to be 50x50 in size and the parameters used for producing the results are $r = 5$, $\delta_I = 0.02$ and $\delta_X = 0.2$. In each set of the images, the left-most one shows the original image. The middle and right panels show the results from two successive bipartitions.

For two of the real images in Figure 5, we compare our method with the normalized cut in the following sub-figures 6(a),6(b),6(c) and 6(d). The first column of each sub-figure shows the first, second and third bipartitions of the images. The second column shows the histogram of the components of the smallest eigenvector, and the right-hand column the distribution of the eigenvector components. The blue and red lines in the right-hand column respectively correspond to zero and the eigenvector component threshold.

Comparing the results of using the commute time and the normalized cut, it is clear that commute time out performs the normalized cut in both maintaining region integrity and continuity. Another important feature is that once again our eigenvector distribution is more stable and discriminates more strongly between clusters.

6 Conclusion

In this paper we have described how commute time can be computed from the Laplacian spectrum. This analysis relies on the discrete Green's function of the graph, and we have reviewed the properties of Green's function. Two of the most important of these are that the Green's function is a kernel and that the commute time is a metric. We show how commute time can be used for clustering and segmentation. Our future plans involve using the commute times to embed the nodes of the graph in a low dimensional space, and to use the characteristics of the embedded node points for the purposes of graph-clustering.

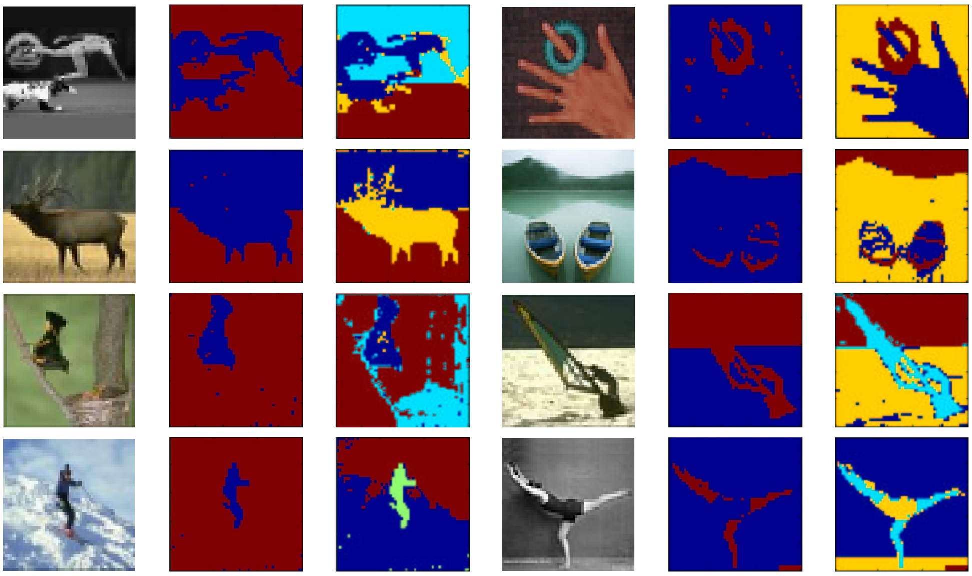


Figure 5: Real world segmentation examples.

References

[1] S. Brin and L.Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[2] F.R.K. Chung. *Spectral Graph Theory*. CBMS series 92. American Mathmatical Society Ed., 1997.

[3] F.R.K. Chung and S.-T. Yau. Discrete green's functions. In *J. Combin. Theory Ser.*, pages 191–214, 2000.

[4] M. Gori, M. Maggini, and L. Sarti. Graph matching using random walks. In *ICPR04*, pages III: 394–397, 2004.

[5] R. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete structures. *19th Intl. Conf. on Machine Learning (ICML) [ICM02].*, 2002.

[6] L. Lovász. Random walks on graphs: A survey.

[7] M. Meila and J. Shi. A random walks view of spectral segmentation, 2001.

[8] M. Pavan and M. Pelillo. A new graph-theoretic approach to clustering and segmentation. In *CVPR03*, pages I: 145–152, 2003.

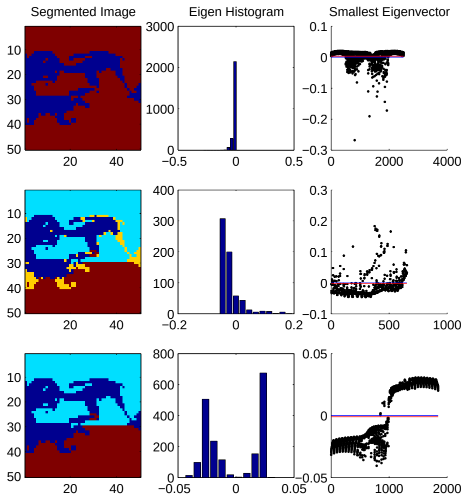
[9] A. Robles-Kelly and E. R. Hancock. String edit distance, random walks and graph matching. *PAMI to appear*, 2005.

[10] M. Saerens, F. Fouss, L. Yen, and P. Dupont. The principal components analysis of a graph, and its relationships to spectral clustering. In *LN-AI*, 2004.

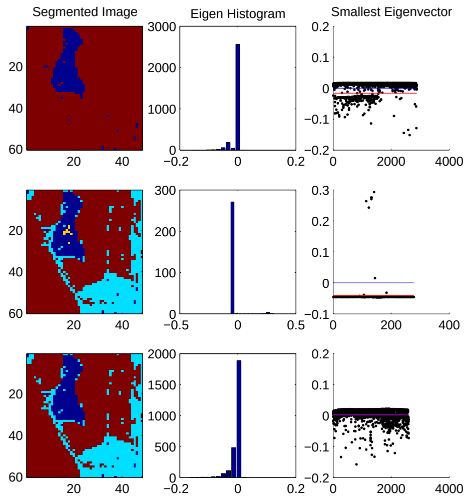
[11] S. Sarkar and K. L. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. In *CVPR*, page 478, 1996.

[12] G. Scott and H. Longuet-Higgins. Feature grouping by relicalisation of eigenvectors of the proximity matrix. In *BMVC.*, pages 103–108, 1990.

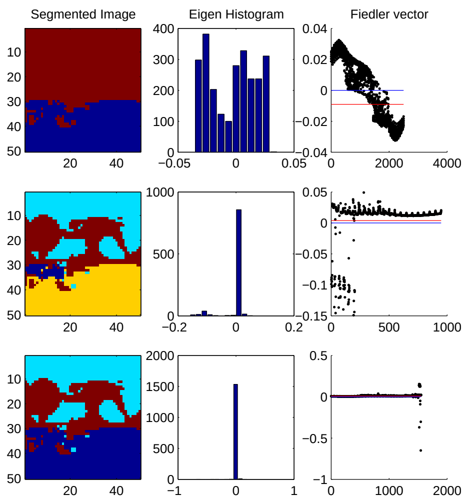
[13] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE PAMI*, 22(8):888–905, 2000.



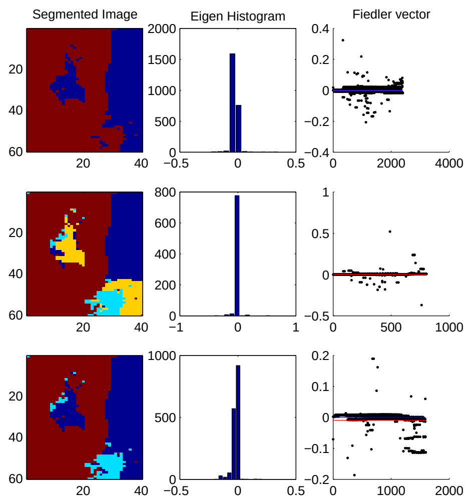
(a) Commute time for 50x50 image with $r = 8$ $\delta_X = 0.5$ $\delta_I = 0.1$



(b) Commute time for 60x40 image with $r = 5$ $\delta_X = 0.2$ $\delta_I = 0.02$



(c) Normalized cut for 50x50 image with $r = 5$ $\delta_X = 2$ $\delta_I = 0.05$



(d) Normalized cut for 60x40 image with $r = 5$ $\delta_X = 0.05$ $\delta_I = 0.01$

Figure 6: Detailed segmentation process in comparasion.

[14] V. Sood, S. Redner, and D. ben Avraham. First-passage properties of the erdoscrenyi random graph. *J. Phys. A: Math. Gen.*, pages 109–123, 2005.