Segmentation using eigenvectors: a unifying view

Yair Weiss
CS Division
UC Berkeley
Berkeley, CA 94720 - 1776
yweiss@cs.berkeley.edu

Abstract

Automatic grouping and segmentation of images remains a challenging problem in computer vision. Recently, a number of authors have demonstrated good performance on this task using methods that are based on eigenvectors of the affinity matrix. These approaches are extremely attractive in that they are based on simple eigendecomposition algorithms whose stability is well understood. Nevertheless, the use of eigendecompositions in the context of segmentation is far from well understood. In this paper we give a unified treatment of these algorithms, and show the close connections between them while highlighting their distinguishing features. We then prove results on eigenvectors of block matrices that allow us to analyze the performance of these algorithms in simple grouping settings. Finally, we use our analysis to motivate a variation on the existing methods that combines aspects from different eigenvector segmentation algorithms. We illustrate our analysis with results on real and synthetic images.

Human perceiving a scene can often easily segment it into coherent segments or groups. There has been a tremendous amount of effort devoted to achieving the same level of performance in computer vision. In many cases, this is done by associating with each pixel a feature vector (e.g. color, motion, texture, position) and using a clustering or grouping algorithm on these feature vectors.

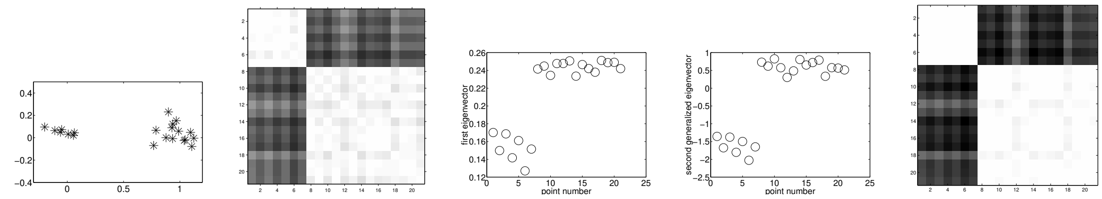
Perhaps the cleanest approach to segmenting points in feature space is based on mixture models in which one assumes the data were generated by multiple processes and estimates the parameters of the processes and the number of components in the mixture. The assignment of points to clusters can then be easily performed by calculating the posterior probability of a point belonging to a cluster. Despite the elegance of this approach, the estimation process leads to a notoriously difficult optimization. The frequently used EM algorithm [3] often converges to a local maximum that depends on the initial conditions.

Recently, a number of authors [11, 10, 8, 9, 2] have suggested alternative segmentation methods that are based on eigenvectors of the (possibly normalized) "affinity matrix". Figure 1a shows two clusters of points and figure 1b shows the affinity matrix defined by:

$$W(i,j) = e^{-d(x_i,x_j)/2\sigma^2} \tag{1}$$

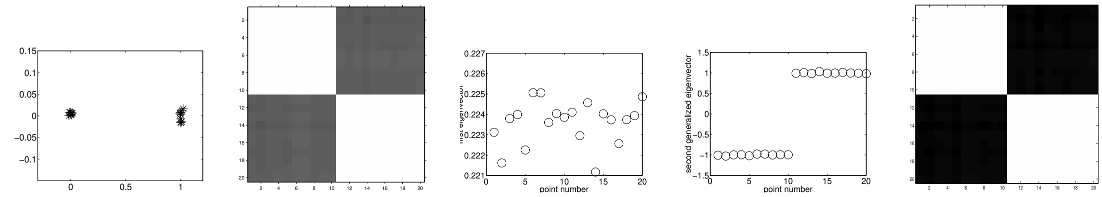
with σ a free parameter. In this case we have used $d(x_i,x_j) = ||x_i - x_j||^2$ but different definition of affinities are possible. The affinities do not even have to obey the metric axioms (e.g. [7]), we will only assume that $d(x_i,x_j) = d(x_j,x_i)$. Note that we have ordered the points so that all points belonging to the first cluster appear first and the points in the second cluster. This helps the visualization of the matrices but does not change the algorithms — eigenvectors of permuted matrices are the permutations of the eigenvectors of the original matrix.

From visual inspection, the affinity matrix contains information about the correct segmentation. In the next section we review four algorithms that look at eigenvectors of affinity matrices. We show that while seemingly quite different, these algorithms are closely related and all use dominant eigenvectors of matrices to perform segmentation. However, these approaches use different matrices, focus on different eigenvectors and use a different method of going from the continuous eigenvectors to the discrete segmentation. In section 2 we prove results on eigendecompositions of block matrices and use these results to analyze the behavior of these algorithms and motivate a new hybrid algorithm. Finally, in section 3 we discuss the application of these algorithms to affinity matrices derived from images.



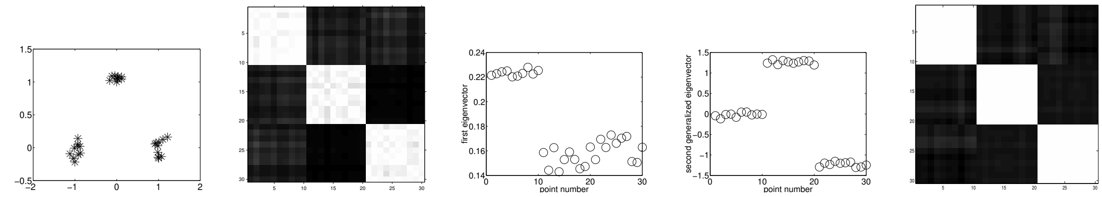
a b c d e

Figure 1: **a.** A simple clustering problem. **b.** The affinity matrix. **c.** The first eigenvector. **d.** The second generalized eigenvector. **e.** The Q matrix.



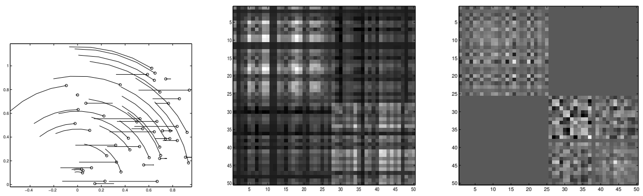
a b c d e

Figure 2: **a.** Another simple clustering problem. **b.** The affinity matrix. **c.** The first eigenvector. **d.** The second generalized eigenvector. **e.** The Q matrix.



a b c d e

Figure 3: **a.** Another simple clustering problem. **b.** The affinity matrix. **c.** The first eigenvector. **d.** The second generalized eigenvector. **e.** The Q matrix.



a b c

Figure 4: **a.** A single frame from a scene with two rigidly moving objects. **b.** The affinity matrix. **c.** The Q matrix.

1 The algorithms

1.1 The Perona and Freeman (1998) algorithm

Perona and Freeman [8] suggested a clustering algorithm based on thresholding the first eigenvector of the affinity matrix (throughout this paper we refer to the "first" eigenvector as the one whose corresponding eigenvalue is largest in magnitude). This is closely related to an approach suggested by Sarkar and Boyer [9] in the context of change detection.

Figure 1c shows the first eigenvector of the affinity matrix in figure 1b. Indeed, the eigenvector can be used to easily separate the two clusters.

Why does this method work? Perona and Freeman have shown that for block diagonal affinity matrices, the first eigenvector will have nonzero components corresponding to points in the dominant cluster and zeros in components corresponding to points outside the dominant cluster. Figure 2 shows that when the nondiagonal blocks are nonzero, the picture is a bit more complicated. Figure 2a shows two very tight clusters where we have constrained both clusters to have exactly the same number of points. Figure 2b shows the affinity matrix with the evident block structure. Figure 2c shows the first eigenvector. Note that there is no correlation between the components of the eigenvalues and the correct segmentation. Figure 3 shows another example where the Perona and Freeman (PF) algorithm works successfully.

1.2 The Shi and Malik (1997) algorithm.

Shi and Malik have argued for using a quite different eigenvector for solving these type of segmentation problems. Rather than examining the first eigenvector of W they look at generalized eigenvectors. Let D be the degree matrix of W:

$$D(i,i) = \sum_j W(i,j) \tag{2}$$

Define the generalized eigenvector y_i as a solution to:

$$(D-W)y_i = \lambda_i D y_i \tag{3}$$

and define the second generalized eigenvector as the y_i corresponding to the second *smallest* λ_i. Shi and Malik suggested thresholding this second generalized eigenvector of W in order to cut the image into two parts. Figure 1c and figure 2c show the second generalized eigenvector of W for the two cases. Indeed these vectors can be easily thresholded to give the correct segmentation.

Why does this method work? Shi and Malik have shown that the second generalized eigenvector is a solution to a continuous version of a *discrete* problem in which the goal is to minimize:

$$\frac{y^T(D-W)y}{y^T D y} \tag{4}$$

subject to the constraint that $y_i \in \{1, -b\}$ and $y^T D 1 = 0$ (where 1 is the vector of all ones).

The significance of the discrete problem is that its solution can be shown to give you the segmentation that minimizes the *normalized cut*:

$$Ncut(A,B) = \frac{cut(A,B)}{asso(A,V)} + \frac{cut(A,B)}{asso(B,V)} \tag{5}$$

where cut(A,B)= $\sum_{i \in A, j \in B} W(i,j)$ and $asso(A,V) = \sum_j \sum_{i \in A} W(i,j)$. Thus the solution to the discrete problem finds a segmentation that minimizes the affinity between groups normalized by the affinity within each group.

As Shi and Malik noted, there is no guarantee that the solution obtained by ignoring the constraints and optimizing equation 4 will bear any relationship to the correct discrete solution. Indeed, they show that the discrete optimization of equation 4 is NP-complete.

Thus the connection to the discrete optimization problem does not rigorously answer the question of why the second generalized eigenvector should give us a good segmentation. Nevertheless, in cases when the solution to the unconstrained problem happens to satisfy the constraints (as in the first two examples), we can infer that it is close to the constrained problems. But what of cases when the second generalized eigenvector doesn't satisfy the constraints? Figure 3a shows an example. The second generalized eigenvector does not have two values but it obviously gives very good information on the correct segmentation (as does the first eigenvector). Why is that?

Note that while Perona and Freeman use the *largest* eigenvector, Shi and Malik use the second *smallest* generalized eigenvector. Thus the two approaches appear quite different. There is, however, a closer connection. Define the normalized affinity matrix:

$$N = D^{-1/2} W D^{-1/2} \tag{6}$$

We call this a *normalized* affinity matrix following [1]. Note that $N(i,j) = W(i,j)/\sqrt{D(i,i)}\sqrt{D(j,j)}$. Given N the following normalization lemma is easily shown:

Normalization Lemma: 1. Let v be an eigenvector of N with eigenvalue λ then $D^{-1/2}v$ is a generalized eigenvector of W with eigenvalue $1-\lambda$. 2. The vector $D^{1/2}1$ is an eigenvector of N with eigenvalue 1.

Thus the second smallest generalized eigenvector of W can be obtained by a componentwise ratio of the second and first *largest* eigenvectors of N. The Shi and Malik (SM) algorithm thus differs from PF in that (1) it uses a *normalized* W matrix and (2) it uses the first two eigenvectors rather than just the first one.

1.3 The Scott and Longuet-Higgins (1990) algorithm.

The Scott and Longuet-Higgins [10] relocalisation algorithm gets as input an affinity matrix W and a number k and outputs a new matrix Q calculated by:

- Constructing the matrix V whose columns are the first k eigenvectors of W.
- normalizing the *rows* of V so that they have unit Euclidean norm. $V(i, \rightarrow) = V(i, \rightarrow)/||V(i, \rightarrow)||$.
- Constructing the matrix $Q = VV^T$.
- Segmenting the points by looking at the elements of Q. Ideally, $Q(i, j) = 1$ if points belong to the same group and $Q(i, j) = 0$ if points belong to different groups.

Figures 1d–3d show the Q matrix computed by the Scott and Longuet-Higgins (SLH) algorithm for the cases surveyed above. Note that in all cases, the $Q(i, j)$ entries for points belonging to the same group are close to 1 and those belonging to different groups are close to 0.

1.4 The Costeira and Kanade (1995) algorithm

Independently of the recent work on using eigenvectors of affinity matrices to segment points in feature space, there has been interest in using singular values of the measurement matrix to segment the points into rigidly moving bodies in 3D [2, 4]. Although these algorithms seem quite different from the ones discussed so far, they are in fact very closely related.

To see the connection, we review the Costeira and Kanade algorithm. Suppose we track n points in f frames. The measurement matrix is a $nx(2f)$ matrix:

$$M = (XY) \tag{7}$$

where $X(i, j), Y(i, j)$ give the x, y coordinate of point i in frame j. The method of Costeira and Kanade segments these points by taking the first k singular vectors of M (where k is the rank of the matrix) and putting them into a matrix V whose columns are the singular vectors. Then constructing the matrix Q by:

$$Q = VV^T \tag{8}$$

Q is a nxn matrix and $Q_{ij} = 0$ for any two points that belong to different objects.

What does this have to do with eigenvectors of affinity matrices? Recall that the singular values of M are by definition the eigenvectors of $W = M^T M$. W is a nxn by matrix that can be thought of as an affinity matrix. The affinity of point i and j is simply the inner product between their traces $(X(i, \rightarrow)Y(i, \rightarrow))$ and $(X(j, \rightarrow)Y(j, \rightarrow))$. Given this definition of affinity, the Costeira and Kanade algorithm is nearly identical to the SLH algorithm. Figure 4 illustrates the Costeira and Kanade algorithm.

2 Analysis of the algorithms in simple grouping settings

In this section we use properties of block matrices to analyze the algorithms. To simplify notation, we assume the data has two clusters. We partition the matrix W into the following form:

$$W = \begin{pmatrix} A & C \\ C^T & B \end{pmatrix} \tag{9}$$

where A and B represent the affinities within the two clusters and C represents the between cluster affinity.

Our strategy in this section is to prove results on idealized block matrices and then appeal to perturbation theorems on eigenvectors [6] to generalize the results to cases where the matrices are only approximately of this form.

2.1 Approximately constant blocks

We begin by assuming the matrices A, B, C are constant. As can be seen from equation 1, this will be the case when the variation of the within and between cluster dissimilarities is significantly smaller than σ. Thus $W(i, j)$ depends only on the membership of points i and j. Note that we do not assume that the between cluster affinity B is zero, or even that it is smaller than the within cluster affinity.

Under these assumptions we can analyze the behavior of the three algorithms exactly:

Claim 1: Assume $W(i, j)$ depends only on the memberships of points i, j. Let v_1 be the indicator vector of the PF algorithm (i.e. the first eigenvector of W) . If point i and point j belong to the same cluster then $v_1(i) = v_1(j)$.

Claim 2: Assume $W(i, j)$ depends only on the memberships of points i, j. Let v be the indicator vector of the SM algorithm (the second generalized eigenvector of W). If point i and point j belong to the same clusters then $v(i) = v(j)$.

Claim 3: Assume $W(i, j)$ depends only on the memberships of points i, j. Let v be the indicator vector of the SM algorithm (the second $Q(i, j)$ in the SLH algorithm with $k = 2$ eigenvectors is equal to

1 if points i and j belong to the same group and 0 otherwise.

The proof of these claims follows from the following decomposition of W:

$$W = OSO^T \tag{10}$$

with O a binary matrix indicating whose columns are membership vectors for the clusters:

$$O = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ \cdots & \cdots \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \tag{11}$$

and S a small $2x2$ matrix that contains the constant values of W:

$$S = \begin{pmatrix} a & c \\ c & b \end{pmatrix} \tag{12}$$

Obviously, if we had an algorithm that given W gave us O then segmentation would be trivial. Unfortunately, the decomposition in equation 10 is not an eigendecomposition so standard linear algebra algorithms will not recover it. However, eigendecomposition algorithms will recover a rotation of a suitably normalized O. It can be shown that if V is a matrix whose two columns are the first two eigenvectors of W then $V = OD_2R$ where D_2 is a $2x2$ diagonal matrix and R is a $2x2$ rotation matrix. Hence the claims.

Note that for the PF and SM algorithms we cannot prove that points belonging to different clusters will have different indicator values. We can only prove that points belonging to same clusters will have the same value. Thus in figure 2c the first eigenvector of W has roughly equal values for all points — both those belonging to the same cluster and those belonging to different clusters. Any visible variation is due to noise. It is only for the SLH algorithm that we can guarantee that points belonging to different clusters will be separated.

2.2 Non-constant block diagonal matrices

Here we assume that the within-cluster affinities, i.e. the matrices A, B are arbitrary matrices with positive elements. The between-cluster affinities, i.e. the matrix C is assumed to be zero. We denote by λ_i^A, λ_i^B the eigenvalues of matrices A and B respectively, ordered by decreasing magnitude.

Claim 4: Assume between cluster affinities are zero and within cluster affinities are positive. Let v_1 be the PF indicator vector. If $\lambda_1^A > \lambda_1^B$ then $v_1(i) > 0$ for all points belonging to the first cluster and $v_1(i) = 0$ for all points belonging to the second cluster.

Claim 5: Assume between cluster affinities are zero and within cluster affinities are positive. Let v be the SM indicator vector then $v(i) = v(j)$ if points i, j belong to the same cluster.

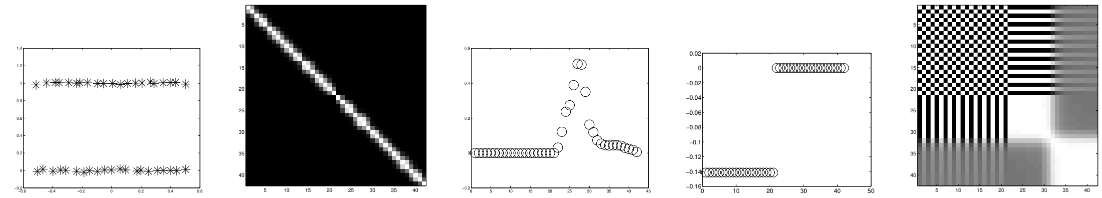
Claim 6: Assume between cluster affinities are zero and within cluster affinities are positive. Let Q be the SLH matrix constructed from W. If $\lambda_1^B > |\lambda_2^A|$ and $\lambda_1^A > |\lambda_2^B|$ then $Q(i,j) = 1$ if i, j belong to the same cluster and zero otherwise.

Claim 4 was proven in [8] and the proof of claim 6 is analogous: if v^a is an eigenvector of A then $v = (v^a; 0)$ is an eigenvector of W with the same eigenvalue. Thus the conditions of claim 6 guarantee that the first two eigenvectors of W will be $(v^a; 0), (0; v^b)$. Claim 5 follows from the normalization lemma proven in the previous section. The vectors $(D_A^{1/2}1; 0)$ and $(0; D_B^{1/2}1)$ are both eigenvectors of N with eigenvalue 1 where D_A, D_B are the degree matrices of A and B. Thus the second generalized eigenvector of W will be some linear combination of these two vectors multiplied by $D^{-1/2}$ so it will be constant for points belonging to the same cluster.

Note that as in the case for constant block matrices, for the PF and SM algorithms we cannot guarantee that points belonging to different clusters can be easily segmented. In the PF algorithm $v(i)$ is guaranteed to be positive for all points in the first cluster, but there is no guarantee of how positive. Figure 5c illustrates this. Many points in the "foreground" cluster have components that are positive yet close to zero. In the SM algorithm, since N has two identical first eigenvalues, v_2 may be any linear combination of the eigenvectors, so the difference between values for the first and second cluster is arbitrary and depends on the implementation details of the eigendecomposition algorithm. In the SLH algorithm, we can again guarantee that different clusters will be segmented but we require an additional constraint on the eigenvalues of the blocks. Figure 5d shows what happens when this additional constraint does not hold. In this case the first two eigenvectors of W are $(0; v_1^b), (0; v_2^b)$ and the Q matrix does not find the correct segmentation.

To summarize, when the matrix has constant blocks then all three algorithms will work, although extracting the discrete segmentation is probably easiest in the SLH algorithm. In this case, normalizing the W matrix does not make any difference. When the blocks are not constant, however, and the between cluster affinities are zeros, the normalization makes a big difference in that it reorders the eigenvectors.

This analysis suggests a combined (SM+SLH) algorithm in which the SLH algorithm is applied to the



a b c d e

Figure 5: **a.** Another simple clustering problem. **b.** The affinity matrix. **c.** The first eigenvector. **d.** The second generalized eigenvector. **e.** The Q matrix of the SLH algorithm.

normalized W matrix, N, rather than to the raw affinity matrix. Indeed, when we run this combined algorithm on the data in figure 5a the correct segmentation is found.

We summarize the properties of the combined (SM+SLH) algorithm:

Claim 7: Assume affinities are only a function of point membership or assume that the between cluster affinities are zero and within cluster affinities are positive. Under both assumptions $Q(i,j)$ in the combined (SM+SLH) algorithm is one if points i and j belong to the same cluster and zero otherwise.

Note that in the idealized cases we have been analyzing, where between cluster affinities are zero and within cluster affinities are positive, then a simple connected-components algorithm will find the correct segmentation. However the perturbation theorems of eigenvectors guarantee that that our claims still hold with small perturbations around these idealized matrices, even when the between cluster affinities are nonzero. In the following section, we show that our analysis for idealized matrices also predicts the behavior on affinity matrices derived from images.

3 Affinity matrices of images

Perona and Freeman conducted a comparison between the first eigenvector of W and the second generalized eigenvector of W when W is constructed by representing each pixel with a (position,intensity) feature vector. In their comparison, the eigenvector of W had a much less crisp representation of the correct segmentation. We have found this to be the case generally for W matrices constructed in this way from images.

Figures 6–9 show examples. Figure 6a shows the baseball player figure from [11]. We constructed a W matrix using the same constants. Figure 6b-e show the first four eigenvectors of W. Note that there is very little information in these eigenvectors regarding the correct segmentation (the pictures do not change when we show log intensities). Figure 6f-i show the first four eigenvectors of the normalized affinity matrix N. Note that at least visually all eigenvectors appear to be correlated with the correct segmentation.

How should this information be recovered? Figure 7a shows the SM indicator vector displayed as an image. Although it contains the information, it is not at all clear how to extract the correct segments from this image — the pixels belonging to the same object do not have constant value but rather have smooth variation. Furthermore, there is obviously additional information in the other eigenvectors.

Figure 7b shows a single column from the matrix Q constructed by the combined (SM+SLH) method with 6 eigenvectors displayed as an image. Ideally, if we had the correct k this column should be all ones for a single object and zeros for points not belonging to the object. Even for k that is too small, this column should have all ones for a single object (but not necessarily zeros for the other pixels). Indeed, we find that the value is nearly one for points belonging to the same object. Figure 7c shows a cross-section. Note that all points corresponding to the baseball player are essentially at 1. It is trivial to extract the baseball player from this representation. Figure 7d show a second column. Again, all pixels corresponding to the second baseball player are very close to 1.

Exactly the same behavior is observed in the dancer image. The information in figure 9a is sufficient to give a segmentation but it is not trivial. In the cross-section (figure 9b) the variation between groups is similar to the variation within groups. Figure 9c-d show the row of the $Q(i,j)$ matrix in the combined (SM+SLH) algorithm and the same cross-section. Extracting the discrete segmentation is trivial.

4 Discussion

Why do eigendecomposition methods for segmentation work? In this paper we have presented a unified view of three of these methods — Perona and Freeman [8], Shi and Malik [11] and Scott and Longuet-Higgins [10]. We showed the similarities and the dif-

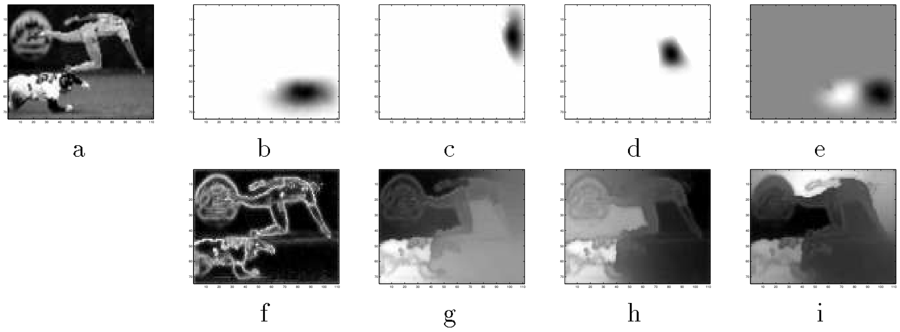


Figure 6: **a.** The baseball image from [11] **b-e.** The eigenvectors of the affinity matrix W. Note there is very little correlation with the desired segmentation. **f-i.** The eigenvectors of the normalized affinity matrix N. Note that all eigenvectors are correlated with the desired segmentation.

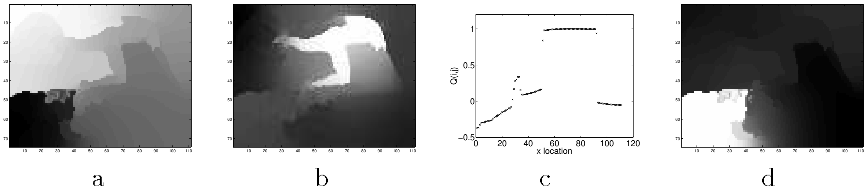


Figure 7: **a.** The second generalized eigenvector of W for the baseball image. Although there is information here regarding the correct segmentation, its extraction is nontrivial. **b.** A row of the Q matrix in the combined (SM+SLH) algorithm for the baseball image. Ideally all pixels corresponding to the same object should have value 1. **c.** A cross section through the pixels in **b.** Note that pixels corresponding to the first baseball player are nearly all 1. **d.** A different row of the Q matrix. All pixels corresponding to the second baseball player are 1.

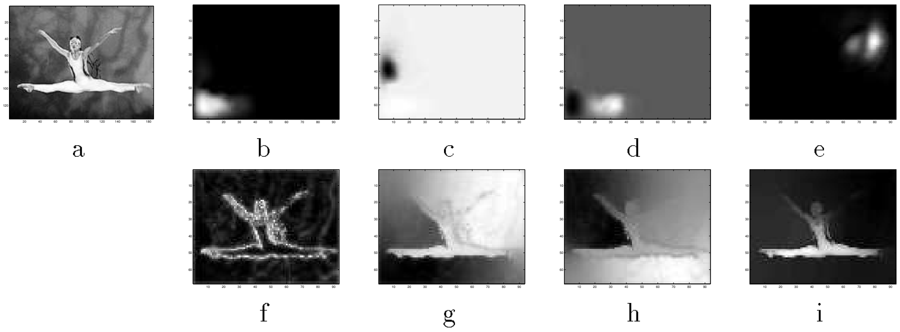


Figure 8: **a.** A gray level image of a ballet dancer. **b-e.** The eigenvectors of the affinity matrix W. Note there is very little correlation with the desired segmentation. **f-i.** The eigenvectors of the normalized affinity matrix N. Note that all eigenvectors are correlated with the desired segmentation.

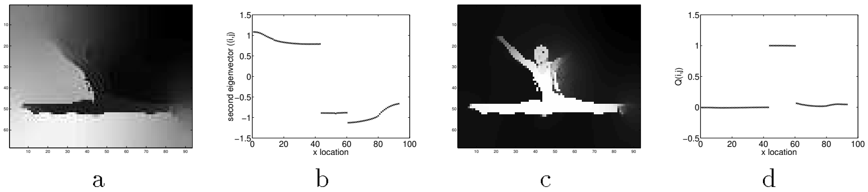


Figure 9: **a.** The second generalized eigenvector of W for the dancer image. Although there is information here regarding the correct segmentation, its extraction is nontrivial. **b.** A horizontal cross section through **a.** Note that the variation between groups is of similar order of magnitude as the variation within groups. **c.** A row of the Q matrix in the combined (SM+SLH) algorithm for the dancer image. Ideally all pixels corresponding to the same object should have value 1. **d.** A cross section through the pixels in **c.** Note that pixels corresponding to the dancer are nearly all 1.

ferences. The similarities are that they all use the top eigenvectors of a matrix. They differ in two ways — which eigenvectors to look at and whether to normalize the W matrix in advance. Using properties of block matrices we showed that when W has constant block structure, all three of these methods will yield eigenvectors that carry some information. We also showed analytically the importance of normalization when the matrix is block diagonal with non-constant blocks. As suggested by the analysis, we found that for real images, unless the W matrix is normalized in the form suggested by Shi and Malik [11] it is nearly impossible to extract segmentation information from the eigenvectors.

In all our analysis and experiments, we never found an example where using normalized W rather than raw W degraded performance. This suggested a scheme that combines the SM algorithm with the SLH algorithm — work with eigenvectors of normalized W but use the first k eigenvectors rather than just the first two. This is similar in spirit to the approach of [12] where the first k eigenvectors of W were used to define a new affinity matrix between the points. Our experimental results on real images are encouraging — by using the first k eigenvectors and combining them into the SLH Q matrix we extract a representation that leads trivially to a discrete segmentation.

We have also discussed a seemingly unrelated rigid body segmentation algorithm — Costeira and Kanade [2] and shown that it is nearly identical to SLH with a particular definition of affinity. It was this connection that motivated the analysis in section 2. We wanted to generalize that type of analysis for arbitrary affinity matrices.

In the case of multibody rigid grouping, there has been additional progress made by using algorithms that do not use eigendecompositions but rather other, more stable matrix decompositions such as the reduced echelon form [4, 5]. Given the close connection between the two problems, we are currently experimenting with using these alternative decompositions in the general grouping context.

The main goal of presenting these algorithms in a unified framework is to enable future work to build on the collective progress made by many researchers in different subfields. We hope that research into the difficult problem of segmentation will benefit from the connections we have pointed out between the different algorithms.

Acknowledgements

I thank W. Freeman, J. Shi, J. Malik and T. Leung for helpful comments and discussions. Supported by MURI-ARO-DAAH04-96-1-0341.

References

[1] F.R.K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[2] J. Costeira and T. Kanade. A multibody factorization method for motion analysis. In *Proc. International Conf. Computer Vision*, pages 1071–1076, 1995.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc. B*, 39:1–38, 1977.

[4] C.W. Gear. feature grouping in moving images. In *Proc IEEE workshop on motion of non-rigid and articulated objects*, pages 214–219, 1994.

[5] C.W. Gear. multibody grouping from motion images. *IJCV*, 29(2):133–150, 1998.

[6] G.H. Golub and C.F. Van-Loan. *Matrix Computations*. Johns Hopkins Press, 1989.

[7] D. W. Jacobs, D. Weinshall, and Y. Gdalyahu. Class representation and image retrieval with non-metric distances. In *Proc. International Conference Computer Vision*, 1998.

[8] P. Perona and W. T. Freeman. A factorization approach to grouping. In H. Burkardt and B. Neumann, editors, *Proc ECCV*, pages 655–670, 1998.

[9] S. Sarkar and K.L. Boyer. quantitative measures of change based on feature organization: eigenvalues and eigenvectors. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1996.

[10] G.L. Scott and H. C. Longuet-Higgins. Feature grouping by relocalisation of eigenvectors of the proxmity matrix. In *Proc. British Machine Vision Conference*, pages 103–108, 1990.

[11] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 731–737, 1997.

[12] J. Shi and J. Malik. Self inducing relational distance and its application to image segmentation. In *Proc. European Conf. Computer Vision*, pages 538–543, 1998.