# Spectral Grouping Using the Nyström Method

Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik

**Abstract**—Spectral graph theoretic methods have recently shown great promise for the problem of image segmentation. However, due to the computational demands of these approaches, applications to large problems such as spatiotemporal data and high resolution imagery have been slow to appear. The contribution of this paper is a method that substantially reduces the computational requirements of grouping algorithms based on spectral partitioning making it feasible to apply them to very large grouping problems. Our approach is based on a technique for the numerical solution of eigenfunction problems known as the Nyström method. This method allows one to extrapolate the complete grouping solution using only a small number of samples. In doing so, we leverage the fact that there are far fewer coherent groups in a scene than pixels.

**Index Terms**—Image and video segmentation, normalized cuts, spectral graph theory, clustering, Nyström approximation.

## 1 Introduction

To humans, an image is more than a collection of pixels; it is a meaningful organization of surfaces and objects in a scene. The Gestalt psychologists were the first to draw attention to this important phenomenon and listed various factors that contribute to this process including grouping cues such as proximity, similarity, and common fate. A great deal of research in computational vision over the last few decades has sought principled ways to operationalize these ideas.

One key component is the development of grouping "engines" that use these low-level cues to perform image and video segmentation. A common characteristic among several recently proposed techniques is the idea of clustering pixels or other image elements using pairwise affinities. The pairwise affinity computed between two pixels captures their degree of similarity as measured by one or more cues. The pixels can then be grouped based on the set of pairwise affinities using methods such as spectral graph partitioning [30], [32], [22], [26], [28], [20], deterministic annealing [25], or stochastic clustering [15].

As discussed in [9], pairwise grouping methods present an appealing alternative to central grouping. Central grouping techniques such as $k$-means or Gaussian Mixture Model fitting via EM [6] tend to be computationally efficient since they only require one to compare the image pixels to a small set of cluster prototypes. However, they have the significant drawback of implicitly assuming that the feature vectors representing the pixels in each group have a Gaussian distribution, justifying the use of Euclidean or Mahalanobis distance for comparing feature vectors. By propagating similarity in a transitive fashion from neighbor to neighbor, pairwise methods can avoid the restriction that all points in a cluster must be close to some prototype. This allows the recovery of clusters that take on more complicated manifold structures in feature space.

Pairwise methods also offer great flexibility in the definition of the affinities between pixels. For example, if the feature vectors represent color histograms, then $k$-means clustering is inappropriate since $L_2$ distance between histograms isn't meaningful. In such a case, pairwise methods can readily employ a suitable affinity function such as the $\chi^2$-distance. Affinities can even be defined between features with no natural vector space structure (e.g., string kernels [17]).

The drawback of pairwise methods is the requirement of comparing all possible pairs of pixels in an image. Processing short video sequences or the output of inexpensive multimegapixel digital cameras can easily involve $10^{12}$ pairwise similarities (a number that will continue to increase in the near future). Consequently, the number of pairs considered in practice is often restricted by placing a threshold on the number of connections per pixel, e.g., by specifying a cutoff radius in the image plane. While this allows the use of efficient sparse representations, it discourages the use of long-range connections, thereby resulting in the oversegmentation of homogeneous regions. In this paper, we present an approximation technique applicable to spectral grouping methods that alleviates this computational burden.

Our approach is based on a classical method for the solution of the integral eigenvalue problem known as the Nyström method. In short, the approximation works by first solving the grouping problem for a small random subset of pixels and then extrapolating this solution to the full set of pixels in the image or image sequence. This provides the flexibility of pairwise grouping with a computational complexity comparable to that of central grouping: Rather than compare all pixels to a set of cluster centers, we

- *C. Fowlkes and J. Malik are with the Electrical Engineering and Computer Science Division, University of California at Berkeley, Berkeley, CA 94720. E-mail: {fowlkes, malik}@cs.berkeley.edu.*
- *S. Belongie is with the Department of Computer Science and Engineering, University of California at San Diego, La Jolla, CA, 92093. E-mail: sjb@cs.ucsd.edu.*
- *F. Chung is with the Department of Mathematics and the Department of Computer Science and Engineering, University of California at San Diego, La Jolla, CA, 92093. E-mail: fan@math.ucsd.edu.*

*Manuscript received 18 Oct. 2002; revised 23 July 2003; accepted 8 Aug. 2003. Recommended for acceptance by W. Freeman.*
*For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117623.*

compare them to a small set of randomly chosen samples. The approach is simple and has the appealing characteristic that for a given number of sample points, its complexity scales linearly with the resolution of the image. In this sense we exploit the fact that the number of coherent groups in an image is generally much smaller than the number of pixels.

The structure of this paper is as follows: In Section 2, we discuss the pairwise grouping framework and review the Normalized Cut [30] grouping algorithm. We discuss the Nyström method in Section 3 and highlight our application to the NCut grouping formulation. We consider the computational costs and approximation error associated with this scheme in Section 4. Results on static images and video are presented in Section 5 and we conclude with Section 6.

## 2 Spectral Methods for Pairwise Clustering

Spectral methods for image segmentation are based on the eigenvectors and eigenvalues of an $N \times N$ matrix derived from the matrix of pairwise affinities. $N$ denotes the number of pixels in the image. These eigenvectors induce an embedding of the pixels in a low-dimensional subspace wherein a simple central clustering method (such as $k$-means) can then be used to do the final partitioning. The spectral method we will focus on in this work is Normalized Cut [30], the background for which is discussed next.[1]

Let the symmetric matrix $W \in \mathbb{R}^{N \times N}$ denote the weighted adjacency matrix for a graph $G = (V, E)$ with nodes $V$ representing pixels and edges $E$ whose weights capture the pairwise affinities between pixels. Let $A$ and $B$ represent a bipartition of $V$, i.e., $A \cup B = V$ and $A \cap B = \emptyset$. Let $\mathrm{cut}(A, B)$ denote the sum of the weights between $A$ and $B$: $\mathrm{cut}(A, B) = \sum_{i \in A, j \in B} W_{ij}$. The degree of the $i$th node is defined as $d_i = \sum_j W_{ij}$ and the volume of a set as the sum of the degrees within that set: $\mathrm{vol}(A) = \sum_{i \in A} d_i$ and $\mathrm{vol}(B) = \sum_{i \in B} d_i$. The Normalized Cut between sets $A$ and $B$ is then given as follows:

$$\mathrm{NCut}(A, B) = \frac{2 \cdot \mathrm{cut}(A, B)}{\mathrm{vol}(A) \| \mathrm{vol}(B)},$$

where $\|$ denotes the harmonic mean.[2]

We wish to find $A$ and $B$ such that $\mathrm{NCut}(A, B)$ is minimized. Appealing to spectral graph theory [11], Shi and Malik [30] showed that an approximate solution may be obtained by thresholding the eigenvector corresponding to the second smallest eigenvalue $\lambda_2$ of the normalized Laplacian $\mathcal{L}$, which is defined as

$$\mathcal{L} = D^{-1/2}(D - W)D^{-1/2} = I - D^{-1/2}WD^{-1/2},$$

where $D$ is the diagonal matrix with entries $D_{ii} = d_i$. The matrix $\mathcal{L}$ is positive semidefinite, even when $W$ is indefinite. Its eigenvalues lie on the interval $[0, 2]$ so the eigenvalues of $D^{-1/2}WD^{-1/2}$ are confined to lie inside $[-1, 1]$.

1. For more detail, readers are referred to [30].
2. Recall $a \| b = 2ab/(a + b)$.

Extensions to multiple groups are possible via recursive bipartitioning or through the use of multiple eigenvectors. In this work, we employ multiple eigenvectors to embed each element into an $N_E$-dimensional Euclidean space, with $N_E \ll N$, such that significant differences in the normalized affinities are preserved while "noise"[3] is suppressed. The $k$-means algorithm is then used to discover groups of pixels in this embedding space.

To find such an embedding, we compute the $N \times N_E$ matrix of the leading eigenvectors $V$ and the $N_E \times N_E$ diagonal matrix of eigenvalues $\Lambda$ of the system

$$(D^{-1/2}WD^{-1/2})V = V\Lambda.$$

The $i$th embedding coordinate of the $j$th pixel is then given by

$$E_{ij} = \frac{V_{i+1,j}}{\sqrt{D_{jj}}}, \quad i = 1, \ldots, N_E, j = 1, \ldots, N,$$

where the eigenvectors have been sorted in ascending order by eigenvalue. Thus, each pixel is associated with a column of $E$ and the final partitioning is accomplished by clustering the columns.

Unfortunately, the need to solve this system presents a serious computational problem. Since $W$ grows as the square of the number of elements in the grouping problem, it quickly becomes infeasible to fit $W$ in memory, let alone compute its leading eigenvectors. One approach to this problem has been to use a sparse, approximate version of $W$ in which each element is connected only to a few of its nearby neighbors in the image plane and all other connections are assumed to be zero [29]. While this makes it possible to employ efficient, sparse eigensolvers (e.g., Lanczos), the effects of this process are not well understood. Our proposed alternative based on sampling allows all affinities to be retained at the expense of some numerical accuracy in their values.

## 3 The Nyström Extension

### 3.1 Background

The Nyström method [21], [3], [23] is a technique for finding numerical approximations to eigenfunction problems of the form

$$\int_a^b W(x, y)\phi(y)dy = \lambda\phi(x).$$

We can approximate this integral equation by evaluating it at a set of evenly spaced points $\xi_1, \xi_2, \ldots \xi_n$ on the interval $[a, b]$ and employing a simple quadrature rule,

$$\frac{(b-a)}{n} \sum_{j=1}^{n} W(x, \xi_j)\hat{\phi}(\xi_j) = \lambda\hat{\phi}(x), \tag{1}$$

where $\hat{\phi}(x)$ is an approximation to the true $\phi(x)$. To solve (1), we set $x = \xi_i$ yielding the system of equations

$$\frac{(b-a)}{n} \sum_{j=1}^{n} W(\xi_i, \xi_j)\hat{\phi}(\xi_j) = \lambda\hat{\phi}(\xi_i) \quad \forall i \in \{1 \ldots n\}.$$

3. For a discussion of denoising in the case of kernel-PCA see [19].

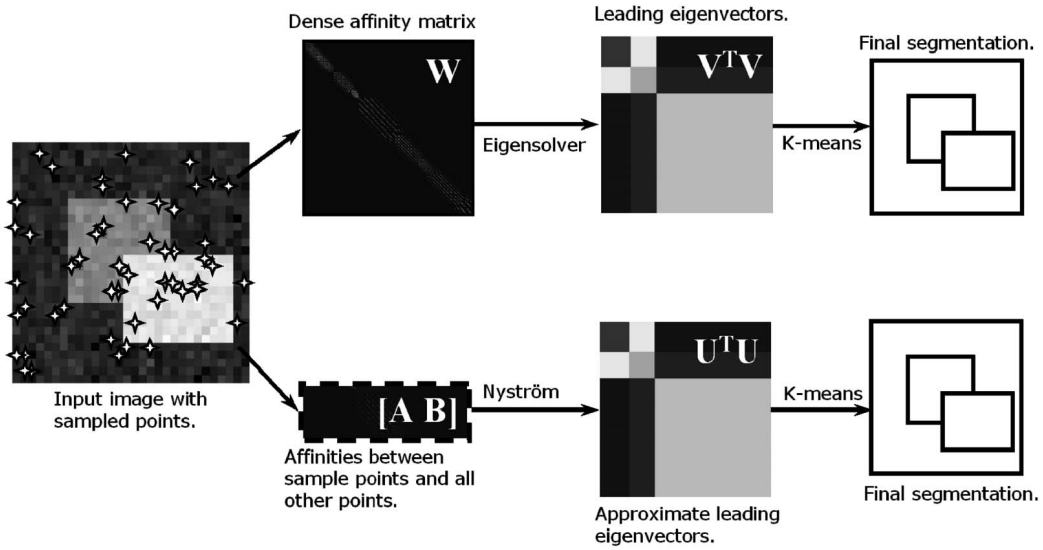


Fig. 1. Flowchart of sampling and matrix completion. At left, a synthetic image is shown consisting of three regions and some additive noise. The dense $N \times N$ affinity matrix $W$, where $N$ is the number of pixels, is shown at top, middle, with entries $W_{ij}$ given by similarity in brightness and position. The entries are ordered so that the pixels in the occluded dark gray square come first, the pixels in the light gray rectangle come next, followed by the pixels from the background. From this dense matrix, one can obtain, at great computational expense, the exact three leading eigenvectors of the normalized Laplacian, denoted $V$. The eigenvectors are illustrated here via their outer product $(V^TV)$ in order to demonstrate their piecewise constant behavior within the three ranges corresponding to the pixels in each group. Using the embedding given by the leading eigenvectors, pixels are clustered with $k$-means to yield a final segmentation. The approximate solution based on the Nyström extension is shown in the lower pathway. Using only those pixels marked by stars on the input image, a narrow strip of the full $W$ matrix is computed, shown at bottom middle. Each row contains the affinities from a sample point to the entire image. The Nyström extension allows one to then directly approximate the leading eigenvectors and segment the image, as shown at bottom right.

Without loss of generality, we let $[a, b]$ be $[0, 1]$ and structure the system as the matrix eigenvalue problem:

$$A\hat{\Phi} = n\hat{\Phi}\Lambda,$$

where $A_{ij} = W(\xi_i, \xi_j)$ and $\Phi = [\phi_1 \phi_2 \ldots \phi_n]$ are the $n$ eigenvectors of $A$ with corresponding eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_n$. Substituting back into (1) yields the *Nyström extension* for each $\hat{\phi}_i$:

$$\hat{\phi}_i(x) = \frac{1}{n\lambda_i} \sum_{j=1}^{n} W(x, \xi_j)\hat{\phi}_i(\xi_j). \tag{2}$$

This expression allows us to extend an eigenvector computed for a set of sample points to an arbitrary point $x$ using $W(\cdot, \xi_j)$ as the interpolation weights.

### 3.2 Matrix Completion

Whereas $x$ in (2) can take on any real value, in the case of image segmentation, the domain over which we wish to extend the solution is specifically those pixels that were not sampled. We can express the evaluation of (2) for those remaining pixels as follows. Let $A$ again be the $n \times n$ matrix of affinities between the sample points with diagonalization $A = U\Lambda U^T$, and let $B$ represent the $n \times m$ matrix of affinities between the $n$ sample points and $m$ remaining points. The matrix form of the Nyström extension is then $B^T U \Lambda^{-1}$, wherein $B^T$ corresponds to $W(\xi_j, \cdot)$, the columns of $U$ correspond to the $\hat{\phi}_i(\xi_j)$s, and $\Lambda^{-1}$ corresponds to the $1/\lambda_i$s in (2). The process is illustrated in schematically in Fig. 1.

To better understand the nature of the Nyström extension, it is instructive to examine it from the standpoint of matrix completion. For simplicity in notation, assume that the $n$ randomly chosen samples come first and the remaining $N - n$ samples come next. Now, partition the affinity matrix $W$ as

$$W = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \tag{3}$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{(N-n) \times n}$, and $C \in \mathbb{R}^{(N-n) \times (N-n)}$. Here, $A$ represents the subblock of weights among the random samples, $B$ contains the weights from the random samples to the rest of the pixels, and $C$ contains the weights between all of the remaining pixels. In the case of interest, $n \ll N$, so $C$ is huge. Letting $\bar{U}$ denote the approximate eigenvectors of $W$, the Nyström extension gives

$$\bar{U} = \begin{bmatrix} U \\ B^T U \Lambda^{-1} \end{bmatrix}$$

and the associated approximation of $W$, which we denote $\hat{W}$, then takes the form

$$\begin{aligned} \hat{W} &= \bar{U}\Lambda\bar{U}^T \\ &= \begin{bmatrix} U \\ B^T U \Lambda^{-1} \end{bmatrix} \Lambda [U^T \quad \Lambda^{-1} U^T B] \\ &= \begin{bmatrix} U\Lambda U^T & B \\ B^T & B^T A^{-1} B \end{bmatrix} \\ &= \begin{bmatrix} A & B \\ B^T & B^T A^{-1} B \end{bmatrix} \\ &= \begin{bmatrix} A \\ B^T \end{bmatrix} A^{-1} [A \quad B]. \end{aligned}$$

Thus, we see that the Nyström extension implicitly approximates $C$ using $B^TA^{-1}B$. The quality of the approximation of the full weight matrix can be quantified as the norm of the Schur complement $\|C - B^TA^{-1}B\|$. The size of this norm is governed by the extent to which $C$ is spanned by the rows of $B$.

The Nyström approximation has been used in this form by [34] for fast approximate Gaussian process classification and regression. As noted in [34], this approximation method directly corresponds to the kernel PCA features space projection technique of [27]. A generalization of these ideas on low-rank approximation to the SVD is studied in [14], [13].

One remaining detail is that the columns of $\bar{U}$ are not orthogonal. The process of orthogonalizing the solution can proceed in two different ways depending on whether $A$ is positive definite.

### 3.3 Methods of Solution

If $A$ is positive definite, then we can solve for the orthogonalized approximate eigenvectors in one step. Let $A^{1/2}$ denote the symmetric positive definite square root of $A$, define $S = A + A^{-1/2}BB^TA^{-1/2}$, and diagonalize it as $S = U_S\Lambda_SU_S^T$. If the matrix $V$ is defined as

$$V = \begin{bmatrix} A \\ B^T \end{bmatrix} A^{-1/2}U_S\Lambda_S^{-1/2}, \tag{4}$$

then one can show (see Appendix A) that $\hat{W}$ is diagonalized by $V$ and $\Lambda_S$, i.e., $\hat{W} = V\Lambda_SV^T$ and $V^TV = I$. We assume that pseudoinverses are used in place of inverses as necessary when there is redundancy in the random samples.

If $A$ is indefinite, then two steps are required to find the orthogonalized solution. Let $\bar{U}_S^T = [U_S^T \quad \Lambda_S^{-1}U_S^TB]$ and define $Z = \bar{U}_S\Lambda^{1/2}$ so that $\hat{W} = ZZ^T$. Let $F\Sigma F^T$ denote the diagonalization of $Z^TZ$. Then, the matrix $V = ZF\Sigma^{-1/2}$ contains the leading orthonormalized eigenvectors of $\hat{W}$, i.e., $\hat{W} = V\Sigma V^T$ with $V^TV = I$. As before, a pseudoinverse can be used in place of a regular inverse when $A$ has linearly dependent columns. Thus, the approximate eigenvectors are produced in two steps: First, we use the Nyström extension to produce $\bar{U}_S$ and $\Lambda_S$ and then we orthogonalize $\bar{U}_S$ to produce $V$ and $\Sigma$. Although this approach is applicable in general, the additional $O(n^3)$ step required leads to an increased loss of significant figures. As noted in [4], it is therefore expedient to know when the one-shot method can be applied, i.e., when a given kernel is positive definite.

### 3.4 Application to Normalized Cut

To apply the Nyström approximation to NCut, it is necessary to compute the row sums of $\hat{W}$. This is possible without explicitly evaluating the $B^TA^{-1}B$ block since

$$\hat{\boldsymbol{d}} = \hat{W}\mathbf{1} = \begin{bmatrix} A\mathbf{1}_n + B\mathbf{1}_m \\ B^T\mathbf{1}_n + B^TA^{-1}B\mathbf{1}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_r + \boldsymbol{b}_r \\ \boldsymbol{b}_c + B^TA^{-1}\boldsymbol{b}_r \end{bmatrix}, \tag{5}$$

where $\boldsymbol{a}_r, \boldsymbol{b}_r \in \mathbb{R}^m$ denote the row sums of $A$ and $B$, respectively, $\boldsymbol{b}_c \in \mathbb{R}^n$ denotes the column sum of $B$, and $\mathbf{1}$ represents a column vector of ones.

With $\hat{\boldsymbol{d}}$ in hand, the required blocks of $\hat{D}^{-1/2}\hat{W}\hat{D}^{-1/2}$ are given by

```
d1 = sum([A;B'],1);
d2 = sum(B,1) + sum(B',1)*pinv(A)*B;
dhat = sqrt(1./[d1 d2])';
A = A.*(dhat(1:n)*dhat(1:n)');
B = B.*(dhat(1:n)*dhat(n+(1:m))');
Asi=sqrtm(pinv(A));
Q=A+Asi*B*B'*Asi;
[U,L,T]=svd(Q);
V=[A;B']*Asi*U*pinv(sqrt(L));
for i = 2:nvec+1
  E(:,i-1) = V(:,i)./V(:,1);
end
```

Fig. 2. Example MATLAB code for finding the first `nvec` embedding vectors of the normalized affinity matrix given unnormalized submatrices `A` of size `n x n` and `B` of size `n x m`. This code uses the "one-shot" technique and so is only applicable to positive definite affinities.

$$A_{ij} \leftarrow \frac{A_{ij}}{\sqrt{\hat{\boldsymbol{d}}_i\hat{\boldsymbol{d}}_j}}, \qquad i,j = 1,\ldots,n$$

and

$$B_{ij} \leftarrow \frac{B_{ij}}{\sqrt{\hat{\boldsymbol{d}}_i\hat{\boldsymbol{d}}_{j+m}}}, \qquad i = 1,\ldots,n, j = 1,\ldots,m$$

to which we can apply one of the two methods from Section 3.3, depending on whether $A$ is positive definite. Fig. 2 gives example MATLAB code for carrying out the computation of the embedding vectors using the "one-shot" technique.

## 4 Performance Considerations

### 4.1 Approximation Properties

It is natural to ask how the approximation actually compares to the solution given by the dense problem or other sparse approximation schemes. In this section we attempt to provide an answer by focusing on an empirical quantitative analysis of performance on a synthetic clustering problem.

The stimulus used for this study consists of the randomly generated annulus/clump pointset shown in Fig. 3a. We increase the difficulty of the grouping task by bringing the clump closer to the annulus; this distance is denoted $R$. The samples are arranged so that the first 50 correspond to the clump and the following 100 correspond to the annulus. The affinities are given by the Gaussian weighted Euclidean distance, i.e., $W_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)$. We measure the quality of the NCut bipartition provided by the second eigenvector in each approximation method using the Fisher criterion [6] which is defined as

$$J(\mathcal{X}_1, \mathcal{X}_2) = \frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2},$$

where $\mu_i$ and $s_i^2$ represent the mean and variance of the points in the $i$th cluster. The parameter $\sigma$ in the affinity function has been chosen to optimize performance as documented in Fig. 3c.

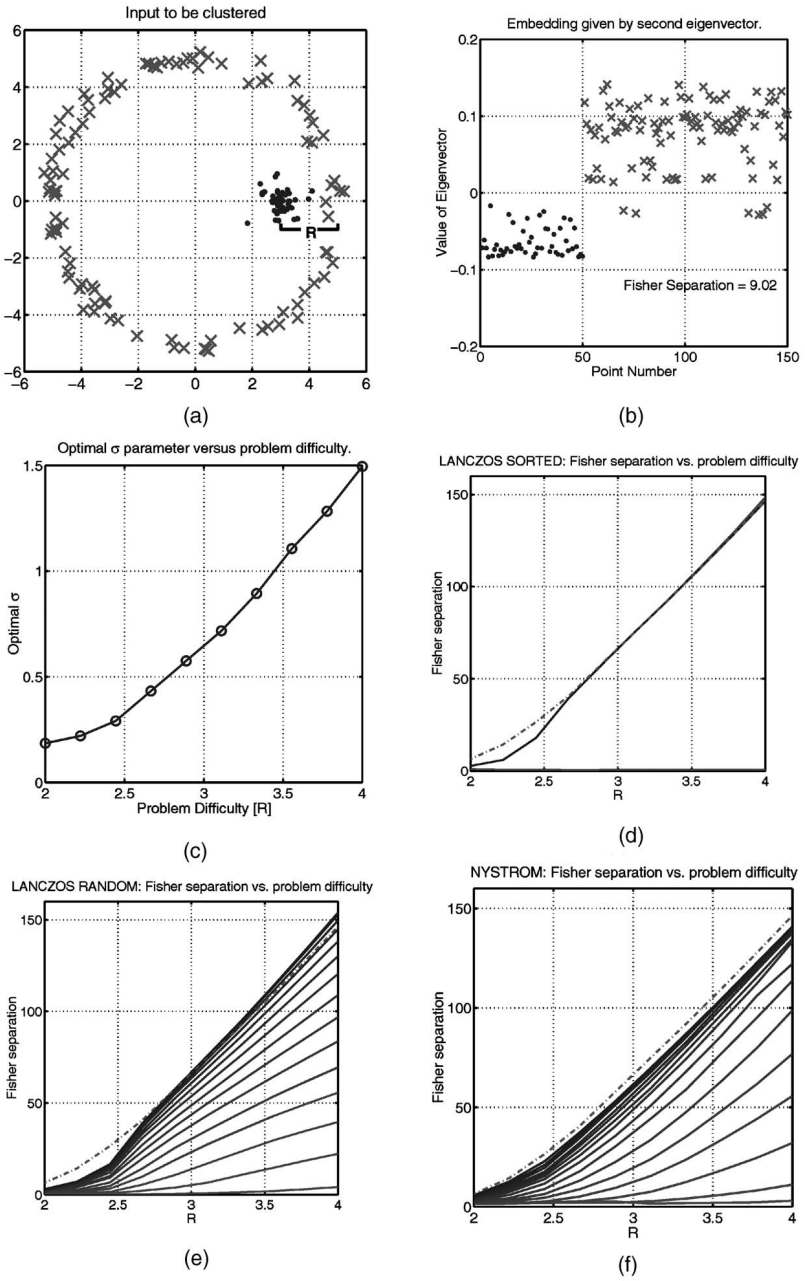


Fig. 3. A study of embedding quality versus number of samples for different approximations. The input stimulus is shown in (a) and a typical embedding given by the second eigenvector of the normalized affinity matrix is shown in (b). The ordering is such that the clump contains points 1-50 and the annulus contains points 51-150. In (c), we show the value of $\sigma$ that optimizes the Fisher separation versus the distance $R$ between the clump and the annulus. The Fisher separation versus problem difficulty for varying numbers of samples is shown in (d), (e), and (f). (d) gives results for sorted Lanczos, (e) for random Lanczos, and (f) for Nyström. The corresponding curve for the dense problem is shown by the dashed line on each plot. Each point on each curve represents the average over 200 random trials. Each solid curve gives the result for a particular number of samples, ranging from 10 to 150; the Fisher separation increases monotonically with the number of samples.

We compare the Nyström approximation to the dense solution along with two other possible approximations based on sparse representations. The first technique is to sort the entries of the affinity matrix and zero out only the smallest ones. For matrices that have many zero or nearly zero entries, this approximation can be quite accurate and preserve exactly the eigenstructure. However, unless there is an oracle that allows one to avoid computing small entries, this still requires $O(N^2)$ affinity calculations which can be quite expensive. A more likely alternative, analyzed in some detail by [1] is to zero out random entries in the matrix. Both of these options allow one to employ a sparse matrix representation and corresponding sparse eigensolver (Lanczos/Arnoldi) which can improve significantly over the $O(N^3)$ complexity required of a dense solution.

The remaining frames in Fig. 3 show the relation between the number of entries used to approximate the eigenvectors of the matrix and the quality of the resulting eigenvectors. Here, the number of samples represents the

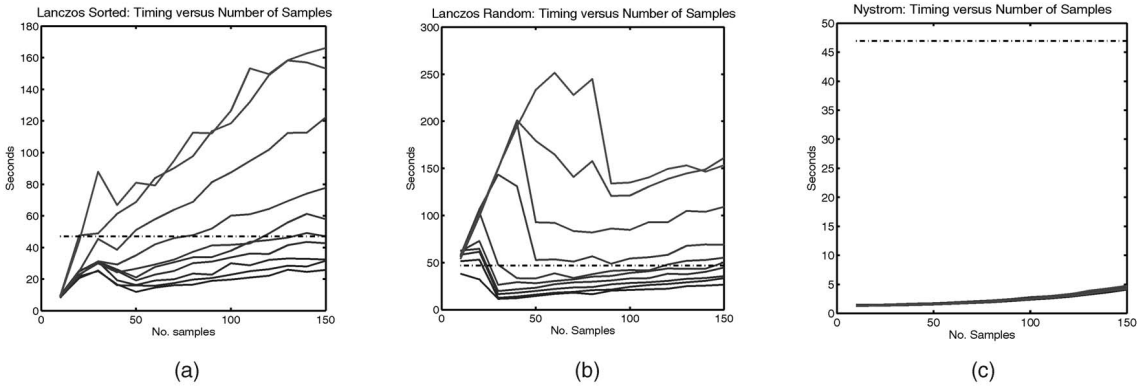


Fig. 4. Running times for approximations versus number of samples at different levels of problem difficulty. (a) shows the sparse sorted Lanczos, (b) shows the randomly sparsified Lanczos method, and (c) shows the Nyström method. The timing for the dense solver is shown by the dotted lines. The timing results are for `svd` and `svds` as implemented by MATLAB. The key thing to notice is that sparse solver performance varies widely with the difficulty (eigenstructure) of the matrix in question. Data here is shown for annulus/clump stimuli consisting of 600 points with timings averaged over 200 trials.

number of nonzero entries above the diagonal (since the matrix is symmetric). This means that each algorithm potentially has access to the same amount of "information" from the affinity matrix.

Fig. 4 makes an empirical comparison of the running times associated with the algorithms. The graphs show the actual running time of a compiled MATLAB implementation versus number of samples. Multiple curves show the timings for increasingly difficult problems (smaller $R$). Asymptotically, the performance of the Lanczos method, $O(n \cdot N \cdot \mathrm{niter})$ operations where niter is the number of iterations to convergence, is quite similar to that of the Nyström technique which takes $O(n^3) + O(n \cdot N)$ operations. However, as the curves in Fig. 4 indicate, while the random Lanczos technique can achieve accuracy similar to that of Nyström given the same number of samples, its running time is highly dependent on the "difficulty" of the problem (highly diagonal matrices take many Lanczos/Arnoldi iterations in practice). In particular, the results in Fig. 4 for $N = 600$ demonstrate that the sparse eigenvector approximation can take longer than simply running MATLAB's dense solver.

### 4.2 Sampling

As suggested above, it is often possible to achieve performance comparable to the dense case using very few samples. We conducted an empirical study to estimate the number of samples needed for a diverse set of natural images. Since it's not possible to solve the dense problem in this case, we use a cross-validation approach. By choosing two different sets of random samples, we can compare the resulting eigenvectors computed by the approximation in order to assess how many samples are necessary for a stable result.

To measure repeatability, we use the Frobenius norm of the inner product $\frac{1}{N_E}\|U^T V\|_{\mathcal{F}}^2$ between sets of leading eigenvectors $U$ and $V$ generated by different random samplings. Note that this measure is only dependent on the subspace spanned by the columns of $U$ and $V$ and hence invariant to rotations of the eigenvectors since for arbitrary rotations $R_1$ and $R_2$

$$\begin{aligned}
&\frac{1}{2N_E}\|UR_1R_1^TU^T - VR_2R_2^TV^T\|_{\mathcal{F}}^2 \\
&= \frac{1}{2N_E}\|UU^T - VV^T\|_{\mathcal{F}}^2 \\
&= \frac{1}{2N_E}\|UU^T\|_{\mathcal{F}}^2 + \frac{1}{2N_E}\|VV^T\|_{\mathcal{F}}^2 - \frac{1}{N_E}\|U^TV\|_{\mathcal{F}}^2 \\
&= 1 - \frac{1}{N_E}\|U^TV\|_{\mathcal{F}}^2.
\end{aligned}$$

For each of 300 images from the Corel data set, we compute 10 different sets of four leading eigenvectors and average the norm between all unique pairs. Fig. 5 shows the result. Perfect agreement would yield a norm of 1 which the approximation quickly converges towards with a small number of samples. The images contain $240 \times 160 = 38,400$ pixels but it's only necessary to sample less than 1 percent of them.

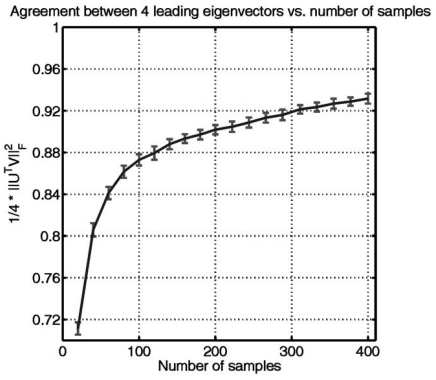


Fig. 5. A cross-validation study of embedding repeatability versus samples for the Nyström approximation based on a set of 300 Corel images of natural scenes, each of size $240 \times 160$. The curve illustrates the agreement in the leading 4 eigenvectors between different random samples of size $n$ ranging from 20 to 400. A norm of 1 indicates perfect agreement. Error bars show the standard deviation over 190 comparisons made between 20 random samplings. These results show that very good agreement in the approximate leading eigenvectors is attained across different random subsets of samples whose size is less than 1 percent of total image pixels.

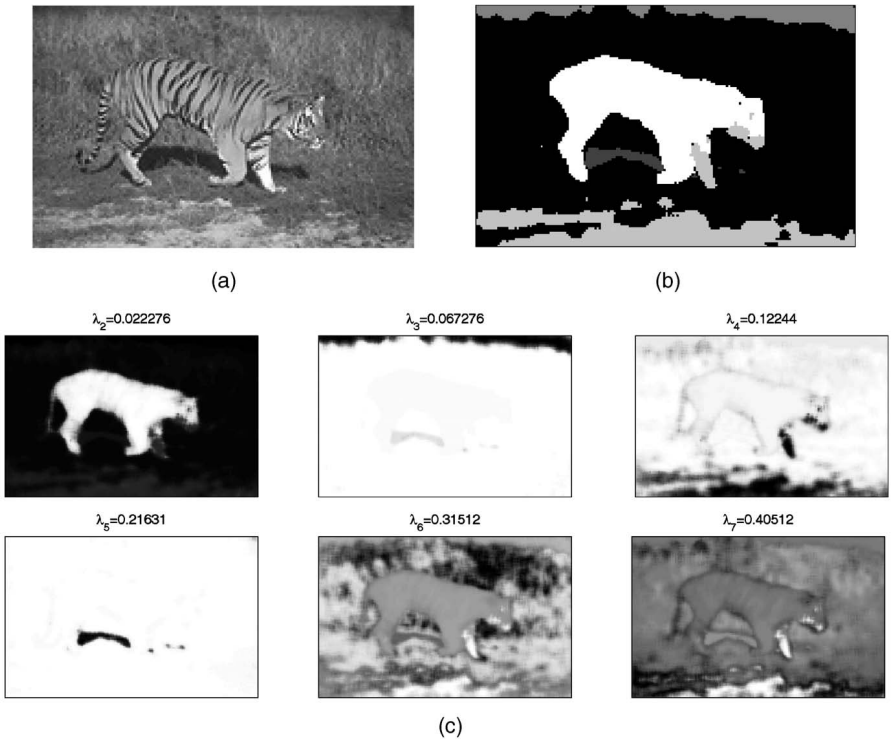


Fig. 6. Segmentation of tiger image based on Gaussian weighted $\chi^2$-distance between local color histograms. The image size is $128 \times 192$ and the histogram window size is $5 \times 5$. Color quantization was performed as in [24] with eight bins. Since the $e^{-\chi^2_{ij}}$ kernel is positive definite, we can use the one-shot method of [12]. (a) Original image. (b) Nyström-NCut leading eigenvectors using 100 random samples. Eigenvector images are sorted by eigenvalue. (c) Segment-label image obtained via $k$-means clustering on the eigenvectors as described in [12].

## 5 Segmentation Results

In this section, we demonstrate the use of the Nyström extension on both static image and video segmentation problems. In each experiment, we used $k$-means with random initialization to cluster the leading $k$ eigenvectors. Choosing $k$ is a difficult model-selection problem which lies outside the scope of this paper. Here, the number of clusters $k$ was chosen manually.

### 5.1 Color and Texture Segmentation

The $\chi^2$ test is a simple and effective means of comparing two histograms. It has been shown to be a very robust measure for color and texture discrimination [25]. Given normalized histograms $h_i(k)$ and $h_j(k)$ define

$$\chi^2_{ij} = \frac{1}{2}\sum_{k=1}^{K}\frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)},$$

where it is understood that a small quantity $\epsilon$ is added to any empty bin so that $h_i(k) > 0 \quad \forall j, k$.

We can then define the similarity between the pair of histograms as $W_{ij} = e^{-\chi^2_{ij}/\alpha}$. Since this kernel is positive definite (see Appendix B) one can employ the one-shot Nyström method to find groups of similar histograms.

An example of Nyström-NCut on a color image of a tiger using 100 samples is shown in Fig. 6. In this example, we computed a local color histogram inside a $5 \times 5$ box around each pixel using the color quantization scheme of [24].

Fig. 7 shows the results of applying Nyström-NCut to texture based segmentation, again using 100 samples. In this case, each pixel in the image is associated with the nearest element in a small alphabet of prototypical linear filter responses using vector-quantitization (see Malik et al. [18]). Histograms of these "texton labels" are computed over an $9 \times 9$ pixel window and again compared with the $\chi^2$-distance.

### 5.2 Spatio-Temporal Segmentation

One method for combining both static image cues and motion information present in a video sequence is to consider the set of images as a space-time volume and attempt to partition this volume into regions that are coherent with respect to the various grouping cues. The insight of considering a video signal as three dimensional for purposes of analysis goes back to Adelson and Bergen [2] and Baker et al. [7] and is supported by evidence from psychophysics [16]. Unified treatment of the spatial and temporal domains is also appealing as it could solve some of the well known problems in grouping schemes based on motion alone (e.g., layered motion models [33], [31]). For example, color or brightness cues can help to segment untextured regions for which the motion cues are ambiguous and contour cues can impose sharp boundaries where optical flow algorithms tend to drag along bits of background regions.

The successes of pairwise grouping have been slow to carry over to the case of spatiotemporal data.[4] Indeed, the conclusions of a recent panel discussion on spatiotemporal grouping [8] are that approaches in which the image sequence is treated as a multidimensional volume in $x, y, t$

4. Some preliminary steps in this direction were made by [29].

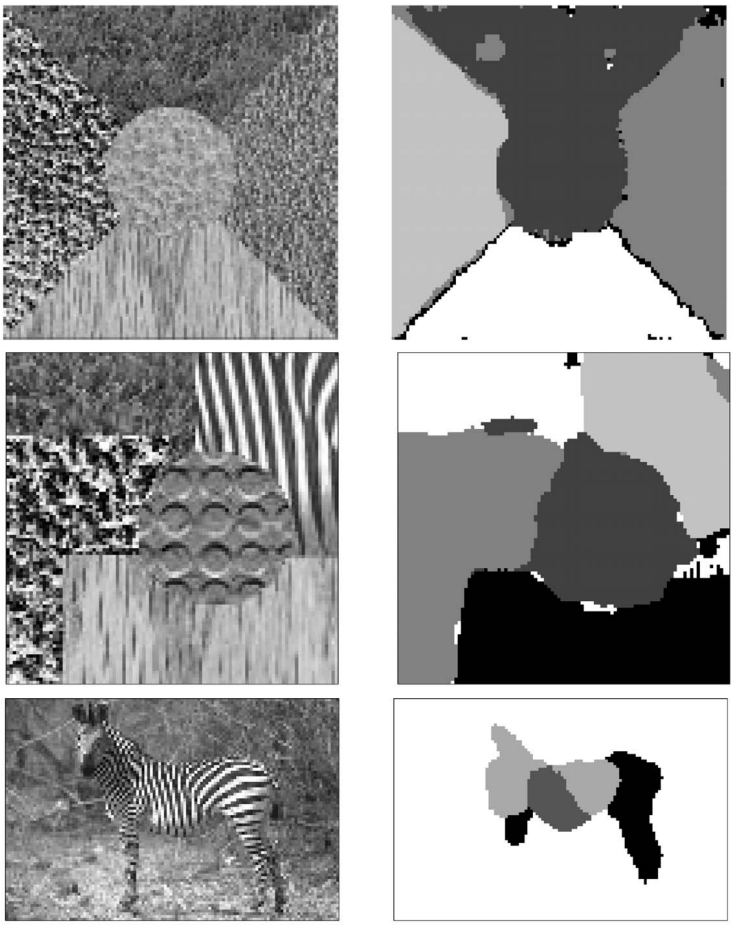

Fig. 7. Segmentation of texture using the Gaussian weighted $\chi^2$-distance between local texton histograms. Images in the left column are given as input. Eigenvectors are computed using 100 random samples and the leading vectors clustered using $k$-means.

hold the greatest promise, but that efforts along these lines have been hampered largely by computational demands. The Nyström approximation has the potential to ameliorate this computational burden, thus making it feasible to extend the ideas of powerful pairwise grouping methods to the domain of video.

We provide two examples of video segmentation using our algorithm. Each of the results shown make use of 100 samples drawn at random from the first, middle and last frame in the sequence. Fig. 8 shows the performance of our algorithm on the flower garden sequence. A proper treatment would require dealing with the texture in the flowerbed and the illusory contours that define the tree trunk. However, the discontinuities in local color and motion alone are enough to yield a fairly satisfying segmentation. Fig. 9 demonstrates segmentation of a relatively uncluttered scene. Processing the entire sequence as a volume automatically provides correspondences between segments in each frame. We note that using motion alone would tend to track the shadows and specularities present on the background and fail to find the sharp boundaries around the body. On a 800MHz Pentium III processor, segmenting a $120 \times 120 \times 5$ voxel sequence takes less than one minute in MATLAB.

## 6 Conclusion

In this paper, we have presented a technique for the approximate solution of spectral partitioning for image and video segmentation based on the Nyström extension. The technique is simple to implement, computationally efficient, numerically stable, and leverages the intuition that the number of groups in an image is generally much smaller than the number of pixels. Our experimental studies on grouping using the cues of texture, color, and optical flow demonstrate that roughly 100 randomly chosen samples are sufficient to capture the salient groups in typical natural images.

## Appendix A
## Proof of One-Shot Method

Suppose that we have

$$W = \begin{bmatrix} A \\ B^T \end{bmatrix} A^{-1} [A \quad B]$$

and we want to show that W can be diagonalized so that

$$W = V \Lambda V^T,$$

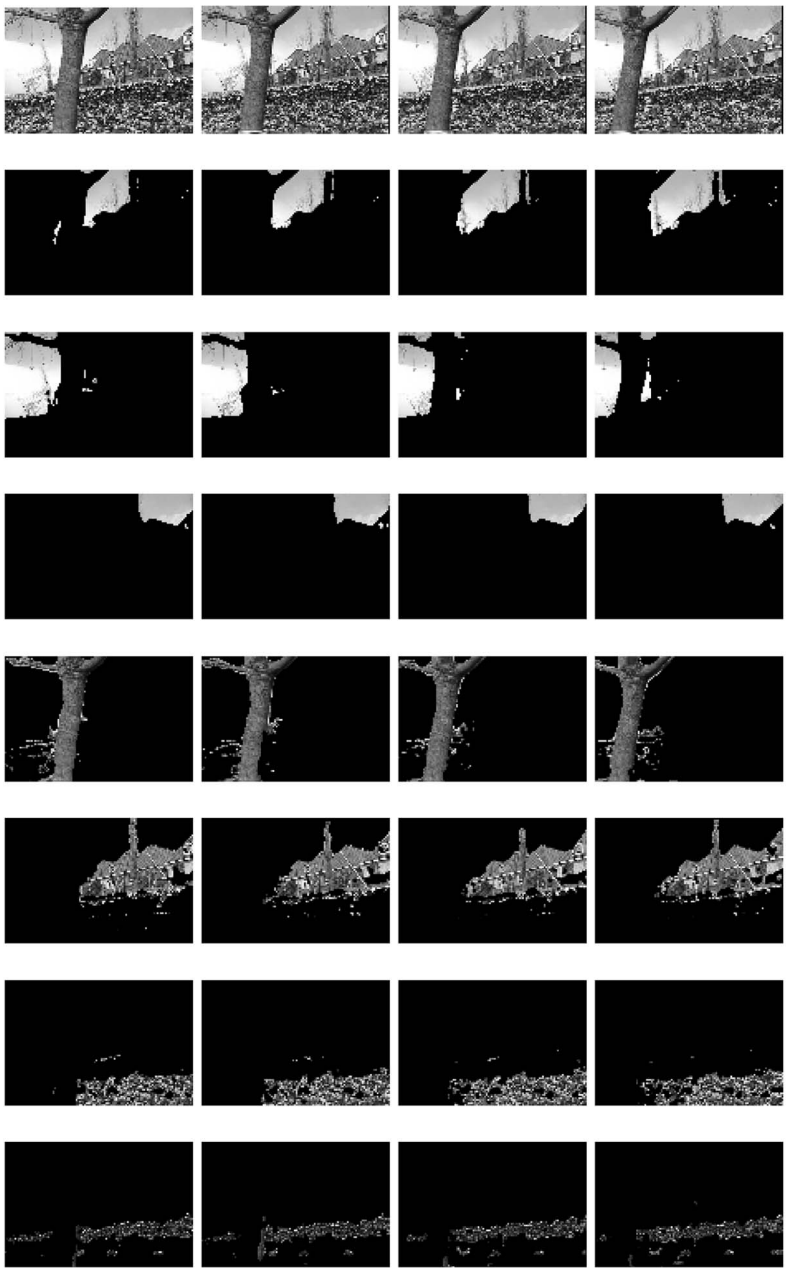

Fig. 8. The flower garden sequence: Each column represents our segmentation of a frame from the sequence of four images shown in the top row. Each row shows slices through a space-time segment. It's important to note that the algorithm provides segment correspondence between frames automatically. The image dimensions are $120 \times 80$ pixels.

where

$$V = \begin{bmatrix} A \\ B^T \end{bmatrix} A^{-1/2} U \Lambda^{-1/2}.$$

To see this, we consider

$$\begin{aligned} W &= \begin{bmatrix} A \\ B^T \end{bmatrix} A^{-1} [A \;\; B] \\ &= \left\{ \begin{bmatrix} A \\ B^T \end{bmatrix} A^{-1/2} U \Lambda^{-1/2} \right\} \Lambda \left\{ \Lambda^{-1/2} U^T A^{-1/2} [A \;\; B] \right\} \\ &= V \Lambda V^T. \end{aligned}$$

The above holds for any diagonal $\Lambda$ and unitary $U$. We wish to determine what they are.

Now, we require

$$\begin{aligned} I &= V^T V \\ &= \left\{ \Lambda^{-1/2} U^T A^{-1/2} [A \;\; B] \right\} \left\{ \begin{bmatrix} A \\ B^T \end{bmatrix} A^{-1/2} U \Lambda^{-1/2} \right\}. \end{aligned}$$

Multiplying from the left by $U\Lambda^{1/2}$ and from the right by $\Lambda^{1/2}U^T$, we have

$$U \Lambda U^T = A^{-1/2} [A \;\; B] \begin{bmatrix} A \\ B^T \end{bmatrix} A^{-1/2} = A + A^{-1/2} B B^T A^{-1/2}.$$

□

Fig. 9. The leap: The original frames ($120 \times 80$ pixels) are shown in the left column. Each column shows slices through a space-time segment.

## APPENDIX B

## PROOF OF POSITIVE DEFINITENESS OF $e^{-\chi^2_{ij}}$

We now prove that $e^{-\chi^2_{ij}}$ is positive definite (as conjectured by [10]). We assume throughout that $h_i(k) > 0 \quad \forall i, k$. We begin by considering the $\chi^2_{ij}$ term by itself. Noting that $(h_i(k) - h_j(k))^2 = (h_i(k) + h_j(k))^2 - 4h_i(k)h_i(k)$, we can rewrite $\chi^2_{ij}$ as

$$\chi^2_{ij} = 1 - 2\sum_{k=1}^{K} \frac{h_i(k)h_j(k)}{h_i(k) + h_j(k)}.$$

We wish to show that the matrix $Q$ with entries given by

$$Q_{ij} = 2\sum_{k=1}^{K} \frac{h_i(k)h_j(k)}{h_i(k) + h_j(k)}$$

is positive definite. Consider the quadratic form $c^T Q c$ for an arbitrary finite nonzero vector $c$:

$$\begin{aligned}
c^T Q c &= \sum_{i,j=1}^{n} c_i c_j Q_{ij} \\
&= 2\sum_{k=1}^{K} \sum_{i,j=1}^{n} c_i c_j \frac{h_i(k)h_j(k)}{h_i(k) + h_j(k)} \\
&= 2\sum_{k=1}^{K} \sum_{i,j=1}^{n} c_i c_j h_i(k)h_j(k) \int_0^1 x^{h_i(k)+h_j(k)-1} \mathrm{dx} \\
&= 2\sum_{k=1}^{K} \sum_{i,j=1}^{n} \int_0^1 c_i h_i(k) x^{h_i(k)-\frac{1}{2}} c_j h_j(k) x^{h_j(k)-\frac{1}{2}} \mathrm{dx} \\
&= 2\sum_{k=1}^{K} \int_0^1 \left( \sum_{i=1}^{n} c_i h_i(k) x^{h_i(k)-\frac{1}{2}} \right) \left( \sum_{j=1}^{n} c_j h_j(k) x^{h_j(k)-\frac{1}{2}} \right) \mathrm{dx} \\
&= 2\sum_{k=1}^{K} \int_0^1 \left( \sum_{i=1}^{n} c_i h_i(k) x^{h_i(k)-\frac{1}{2}} \right)^2 \mathrm{dx} \\
&> 0.
\end{aligned}$$

Thus, $Q$ is positive definite.

Returning now to $e^{-\chi^2_{ij}}$, we note that it can be written as a positive constant times $e^{Q_{ij}}$. Since the exponential of a

positive definite function is also positive definite [5], we have established that $e^{-\chi^2_{ij}}$ is positive definite. □

## ACKNOWLEDGMENTS

The authors would like to thank Gianluca Donato, William Kahan, Andrew Ng, Jianbo Shi, Yair Weiss, Stella Yu, and Alice Zheng for helpful discussions during the course of this work. They would also like to thank Olivier Chapelle for pointing out an error in the original proof in Appendix B and Joachim Buhmann for suggesting the use of cross-validation in Section 4.2.

## REFERENCES

[1] D. Achlioptas, F. McSherry, and B. Schölkopf, "Sampling Techniques for Kernel Methods," *Proc. Neural Information Processing Systems Conf.,* pp. 335-342, 2002.
[2] E.H. Adelson and J.R. Bergen, "Spatiotemporal Energy Models for the Perception of Motion," *J. Optical Society Am. A,* vol. 2, no. 2, pp. 284-299, 1985.
[3] C.T.H. Baker, *The Numerical Treatment of Integral Equations.* Oxford: Clarendon Press, 1977.
[4] S. Belongie, C. Fowlkes, F. Chung, and J. Malik, "Spectral Partitioning with Indefinite Kernels Using the Nyström Extension," *Proc. European Conf. Computer Vision,* 2002.
[5] C. Berg, J.P.R. Christensen, and P. Ressel, *Harmonic Analysis on Semigroups.* Springer-Verlag, 1984.
[6] C.M. Bishop, *Neural Networks for Pattern Recognition.* Oxford Univ. Press, 1995.
[7] R.C. Bolles, H.H. Baker, and D.H. Marimont, "Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion," *Int'l. J. Computer Vision,* vol. 1, pp. 7-55, 1987.
[8] K. Boyer, D. Fagerström, M. Kubovy, P. Johansen, and S. Sarkar, "POCV99 Breakout Session Report: Spatiotemporal Grouping," *Perceptual Organization for Artificial Vision Systems,* S. Sarkar and K.L. Boyer, eds., 2000.
[9] J.M. Buhmann, "Data Clustering and Learning," *The Handbook of Brain Theory and Neural Networks,* M.A. Arbib, ed., pp. 278-281, 1995.
[10] O. Chapelle, P. Haffner, and V. Vapnik, "SVMs for Histogram Based Image Classification," *IEEE Trans. Neural Networks,* vol. 10, no. 5, pp. 1055-1064, Sept. 1999.
[11] F.R.K. Chung, *Spectral Graph Theory.* Am. Math. Soc., 1997.
[12] C. Fowlkes, S. Belongie, and J. Malik, "Efficient Spatiotemporal Grouping Using the Nyström Method," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* Dec. 2001.
[13] A. Frieze and R. Kannan, "Quick Approximation to Matricies and Applications," *Combinatorica,* vol. 19, pp. 175-220, 1999.
[14] A. Frieze, R. Kannan, and S. Vempala, "Fast Monte-Carlo Algorithms for Finding Low-Rank Approximations," *Proc. IEEE Symp. Foundations of Computer Science,* pp. 370-378, 1998.
[15] Y. Gdalyahu, D. Weinshall, and M. Werman, "Stochastic Image Segmentation by Typical Cuts," *Proc. Conf. Computer Vision and Pattern Recognition,* 1999.
[16] S. Gepshtein and M. Kubovy, "The emergence of visual objects in space-time," *Nat'l Academy of Science USA,* vol. 97, no. 14, pp. 8186-8191, 2000.
[17] D. Haussler, "Convolution Kernels on Discrete Structure," technical report, Univ. of California at Santa Cruz, 1999.
[18] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and Texture Analysis for Image Segmentation," *Int'l. J. Computer Vision,* vol. 43, no. 1, pp. 7-27, June 2001.
[19] S. Mika, B. Schölkopf, A.J. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and De-Noising in Feature Spaces," *Advances in Neural Information Processing Systems 11,* pp. 536-542, 1999.
[20] A.Y. Ng, M.I. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an Algorithm," *Proc. Neural Information Processing Systems Conf.,* 2002.
[21] E.J. Nyström, "Über die Praktische Auflösung von Linearen Integralgleichungen mit Anwendungen auf Randwertaufgaben der Potentialtheorie," *Commentationes Physico-Mathematicae,* vol. 4, no. 15, pp. 1-52, 1928.
[22] P. Perona and W.T. Freeman, "A Factorization Approach to Grouping," *Proc. Fifth European Conf. Computer Vision,* 1998.
[23] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical Recipies in C,* second ed. Cambridge Univ. Press, 1992.
[24] J. Puzicha and S. Belongie, "Model-Based Halftoning for Color Image Segmentation," *Proc. Int'l Conf. Pattern Recognition,* vol. 3, pp. 629-632, 2000.
[25] J. Puzicha, T. Hofmann, and J. Buhmann, "Non-Parametric Similarity Measures for Unsupervised Texture Segmentation and Image Retrieval," *Computer Vision and Pattern Recognition,* 1997.
[26] S. Sarkar and K.L. Boyer, "Quantitative Measures of Change Based on Feature Organization: Eigenvalues and Eigenvectors," *Proc. Conf. Computer Vision and Pattern Recognition,* 1996.
[27] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation,* vol. 10, pp. 1299-1319, 1998.
[28] G.L. Scott and H.C. Longuet-Higgins, "An Algorithm for Associating the Features of Two Images," *Proc. Royal Soc. London,* vol. B-244, pp. 21-26, 1991.
[29] J. Shi and J. Malik, "Motion Segmentation and Tracking Using Normalized Cuts," *Proc. Int'l Conf. Computer Vision,* Jan. 1998.
[30] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.
[31] Y. Weiss, "Smoothness in Layers: Motion Segmentation Using Nonparametric Mixture Estimation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 520-526, 1997.
[32] Y. Weiss, "Segmentation Using Eigenvectors: A Unifying View," *Proc. Seventh Int'l. Conf. Computer Vision,* pp. 975-982, 1999.
[33] Y. Weiss and E. Adelson, "A Unified Mixture Framework for Motion Segmentation: Incorporating Spatial Coherence and Estimating the Number of Models," *Proc Conf. Computer Vision and Pattern Recognition,* pp. 321-326, June 1996.
[34] C. Williams and M. Seeger, "Using the Nyström Method to Speed Up Kernel Machines," *Advances in Neural Information Processing Systems 13: Proc. 2000 Conf.,* T.K. Leen, T.G. Dietterich, and V. Tresp, eds., pp. 682-688, 2001.

**Charless Fowlkes** received the BS degree with honors in engineering and applied sciences from the California Institute of Technology in 2000. He is a PhD student at the University of California at Berkeley in the Computer Science Division, where his research has been supported by a UC MICRO fellowship and by a US National Science Foundation Graduate research fellowship. His research interests lie in the ecological statistics of natural scenes, perceptual organization, and machine learning.

**Serge Belongie** received the BS degree (with honor) in electrical engineering from the California Institute of Technology in 1995 and the MS and PhD degrees in electrical engineering and computer sciences (EECS) from the University of California at Berkeley in 1997 and 2000, respectively. While at Berkeley, his research was supported by a the US National Science Foundation Graduate research fellowship and the Chancellor's Opportunity Predoctoral fellowship. He is also a cofounder of Digital Persona, Inc., and the principal architect of the Digital Persona fingerprint recognition algorithm. He is currently an assistant professor in the Computer Science and Engineering Department at the University of California at San Diego. His research interests include computer vision, pattern recognition, and digital signal processing.

**Fan Chung** is the Akamai professor of internet mathematics at the Univ. of California at San Diego. Her main research interests lie in spectral graph theory and extremal graph theory. She has published more than 200 papers, in areas ranging from pure mathematics (e.g., differential geometry, number theory) to the applied (e.g., optimization, computational geometry, telecommunications, and Internet computing). She has about 100 coauthors including many mathematicians, computer scientists, statisticians, and chemists. She is the author of two books—Erdös on Graphs and Spectral Graph Theory. She is the editor-in-chief of a new journal, *Internet Mathematics*, and the coeditor-in-chief of *Advances in Applied Mathematics*. In addition, she serves on the editorial boards of a dozen or so journals. She was awarded the Allendoerfer Award from the Mathematical Association of America in 1990. Since 1998, she has been a fellow of American Academy of Arts and Sciences.

**Jitendra Malik** received the BTech degree in electrical engineering from Indian Institute of Technology, Kanpur, in 1980 and the PhD degree in computer science from Stanford University in 1985. In January 1986, he joined the University of California at Berkeley, where he is currently the Arthur J. Chick Endowed Professor of electrical engineering and computer science, and the associate chair for the Computer Science Division. He is also on the faculty of the Cognitive Science and Vision Science Groups. His research interests are in computer vision and computational modeling of human vision. His work spans a range of topics in vision including image segmentation and grouping, texture, stereopsis, object recognition, image based modeling and rendering, content based image querying, and intelligent vehicle highway systems. He has authored or coauthored more than a hundred research papers on these topics. He received the gold medal for the best graduating student in electrical engineering from IIT Kanpur in 1980, a Presidential Young Investigator Award in 1989, and the Rosenbaum fellowship for the Computer Vision Programme at the Newton Institute of Mathematical Sciences, University of Cambridge in 1993. He received the Diane S. McEntyre Award for excellence in teaching from the Computer Science Division, University of California at Berkeley, in 2000. He was awarded a Miller Research Professorship in 2001. He is an editor-in-chief of the *International Journal of Computer Vision*, and an associate editor of the *Journal of Vision*. He also serves on the scientific advisory board of the Pacific Institute for the Mathematical Sciences.