# SPECTRAL CLUSTERING WITH MEAN SHIFT PREPROCESSING

Umut Ozertem, Deniz Erdogmus
CSEE Department, OGI, Oregon Health & Science University, Portland, Oregon, USA

**Abstract.** Clustering is a fundamental problem in machine learning with numerous important applications in statistical signal processing, pattern recognition, and computer vision, where unsupervised analysis of data classification structures are required. The current state-of-the-art in clustering is widely accepted to be the so-called spectral clustering. Spectral clustering, based on pairwise affinities of samples imposes very large computational requirements. In this paper, we propose a vector quantization preprocessing stage for spectral clustering similar to the classical mean-shift principle for clustering. This preprocessing reduces the dimensionality of the matrix on which spectral techniques will be applied, resulting in significant computational savings.

## 1. INTRODUCTION

Data clustering is an important fundamental problem having a wide range of applications on different aspects of unsupervised learning; image segmentation, data mining, speech recognition, and data compression to name just a few. In recent years there has been a growing interest on spectral clustering and it is recognized as an important tool for clustering problems. In spectral clustering, data segmentation is obtained using the eigendecomposition of an affinity matrix that defines the similarities in the data. In the definition of the affinity matrix, different similarity measures can be utilized to characterize the affinities. The affinities do not even have to obey the metric axioms except the symmetry property.

Spectral clustering dates back to the discovery of the utilization of the second eigenvector of the Laplacian matrix to bi-partition the data, by Fiedler [1]. Recently, a number of related clustering methods are suggested that utilize the eigenvectors or generalized eigenvectors of the affinity matrix [2-14]. Such methods are known as spectral clustering and considered to be the state-of-the art clustering methods in the literature.

The majority of the spectral clustering algorithms are different variants of graph cut and multiway cut methods, each using different affinity matrices and utilizing the resulting eigendecomposition in different manners. Obtaining the eigendecomposition, the clustering is obtained by thresholding the values of a suitably selected eigenvector. One should also notice that these methods are sensitive to the definition of affinity between the data pairs, and since no theoretical criterion for choosing the functions to assign the affinities are known, these algorithms require the assumption of the existence of a suitable affinity definition.

A different track in spectral clustering was designated by Scott and Longuet-Higgins [12], in which they propose a mapping using the eigenvectors of the affinity matrix to transform the data from the original data space to the kernel induced feature space, and do the actual clustering on the image of the data in that space. Normalization of the transformed data is an important step in this approach, and provided that, clustering of the image of the data in the kernel induced feature space is shown to be generating very successful results for a variety of different data sets. Spectral clustering can be understood as measuring sample similarities by an inner product in the kernel-induced feature space. Using Mercer kernels, the *kernel trick* defines a technique to compute the inner products in the potentially infinite dimensional kernel induced feature space. Kernel-based methods rely on the assumption that the clustering in the kernel induced feature space is easier compared to the original clustering problem. In practice, one cannot prove that this assumption holds for all Mercer kernels, on the other hand, one could search for a kernel that makes this desired property true. Kernel optimization, in general, is a daunting task and there are no practical solutions yet. We will exploit the connection of kernel methods with kernel density estimation to be able to utilize well-known results from this literature to select an appropriate kernel [15].

The main shortcoming of the spectral clustering algorithms is the computational complexity, since these algorithms require the computation of the eigenvectors of the $N \times N$ affinity matrix, where $N$ is the sample size. The computational complexity of all the eigenvector calculations is $O(N^3)$, which makes the spectral clustering methods impractical to use for large data sets.

In this paper, we propose a spectral clustering algorithm using fixed-size kernel density estimation with a mean shift algorithm to represent the data in a much smaller and quantized affinity matrix. This leads to a reduced computational complexity, which is defined in the order of modes present in the data probability density function.

## 2. THE PROPOSED METHOD

We discuss the details of the proposed method in this section after a brief overview of spectral clustering. Given a set of vectors $\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$ and a *suitable* kernel function $K_\sigma(\mathbf{x}_i,\mathbf{x}_j)$ — the measure of pairwise affinity or closeness — where $\sigma$ denotes the kernel size (e.g., the standard deviation in the case of a Gaussian kernel), the affinity matrix $\mathbf{K}$ or the normalized graph Laplacian matrix $\mathbf{L}$ are constructed as shown in (1) [5,7,8].

$$\begin{aligned}\mathbf{K}_{ij} &= K_\sigma(\mathbf{x}_i, \mathbf{x}_j) \\ \mathbf{L}_{ij} &= D_i^{-1/2}\mathbf{K}_{ij}D_j^{-1/2}\end{aligned} \tag{1}$$

where the normalization terms are given by $D_i = \sum_j \mathbf{K}_{ij}$.

The clustering solution is achieved using one of the following three approaches on these matrices (the normalized Laplacian is usually the matrix of choice due to its improved eigenspread [8] and relationship with normalized graph cuts [5,6,11]):

1. Threshold the largest eigenvector of **K** [7].
2. Threshold the second largest eigenvector of **L** [5].
3. Transform the data to the kernel-induced feature space using the eigenvectors of **K** or **L** and use a simple clustering algorithm in that domain [12].

The size of both **K** and **L** is $N$×$N$; therefore, the calculation of necessary eigenvectors becomes cumbersome for very large $N$.

Recently, it has been shown that spectral clustering approaches based on the affinity and Laplacian matrices are intrinsically related to kernel density estimation and assignment of cluster labels to minimize between-cluster overlap and within cluster entropy [16]. Specifically, using a fixed-size kernel density estimate, spectral clustering is cast as an optimisation problem where the *angle* between the cluster distributions is to be maximized as measured by the inner product between the tentative cluster distributions at any step of label assignment iterations.

Motivated by this relationship between spectral clustering and fixed-size kernel density estimation, we propose a two-stage nonparametric clustering algorithm that combines the mean-shift principle [17,18] and spectral clustering technique. The first stage of this approach aims to determine the modes of the nonparametric kernel density estimate of the data, which acts as a vector quantization preprocessing stage for the following spectral clustering stage. This procedure reduces the dimensionality of the affinity/Laplacian matrix from $N$ to $M$, where $M$ is the number of modes of the estimated data distribution as determined by the mean-shift procedure. Typically, $M$<<$N$, resulting in significant savings in spectral computations. In the following, we focus on the 2-cluster case for simplicity. The technique, however, generalizes to arbitrary number of clusters easily, assuming that the number of clusters can be either estimated correctly or input to the system and leaving the estimation of number of clusters as a future work.

### 2.1. Decision Boundary for Clustering

In classification, given the knowledge of the true underlying class distributions $q_1(\mathbf{x})$ and $q_2(\mathbf{x})$ and their corresponding a priori probabilities $p_1$ and $p_2$, the optimal Bayes classifier that minimizes the average probability of error can be easily determined and the corresponding separation boundary is given by the solution to the equation $p_1q_1(\mathbf{x})=p_2q_2(\mathbf{x})$. In clustering, we do not have access to the individual cluster distributions; however, assume that the overall data distribution is known to be $q(\mathbf{x})=p_1q_1(\mathbf{x})+p_2q_2(\mathbf{x})$, where $q_i(\mathbf{x})$ are unimodal distributions for the sake of discussion simplicity. Given $q(\mathbf{x})$, a reasonable clustering boundary is the local minima between the modes corresponding to different clusters, on which the gradient is zero, i.e., $\nabla q(\mathbf{x})=\mathbf{0}$. To illustrate this, we present in Fig. 1 two cases involving two Gaussian classes with balanced and unbalanced a priori probabilities. The minimum of the overall distribution between the clusters is a reasonable approximation to the optimal Bayes boundary in both cases (due to the symmetry of the density values and their gradients around the boundary).

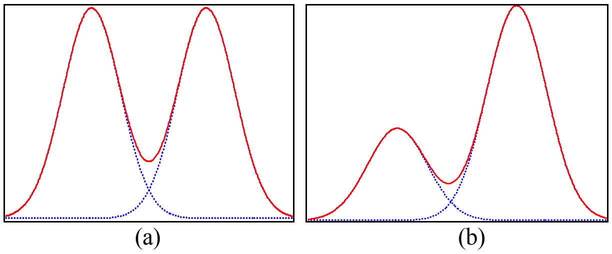
(a) (b)

Figure 1. Two Gaussian clusters with (a) balanced and (b) unbalanced a priori probabilities. The boundary based on the overall data distribution (solid) determined by solving the zero-gradient equation to find the local minimum density point between the clusters is a good approximation of the optimal Bayes boundary based on the individual weighted cluster distributions (dotted) determined by the intersection point.

### 2.2. Nonparametric Density Estimation

In practice, the analytical expression for the data distribution is generally unknown. Furthermore, in many applications these distributions take complex forms and determining suitable parametric families without compromising model accuracy may not be trivial. Nonparametric density estimation techniques alleviate this difficulty by offering versatile alternatives. In this paper, we specifically employ the kernel density estimation technique [19,20] for mainly two reasons: (i) it is a flexible and established nonparametric density estimation that leads to continuous and smooth density estimates, which makes local update based search algorithms feasible, (ii) it allows a natural connection with state-of-the-art spectral clustering approaches, thus naturally allows the construction of a pairwise affinity matrix between the determined modes for spectral analysis in the second stage.

Given a set of samples $\{\mathbf{x}_1,\ldots,\mathbf{x}_N\}$ and a kernel function $K_\sigma(.)$, where $\sigma$ denotes the kernel size (we assume spherical symmetry for simplicity, but this assumption can be easily relaxed), the kernel estimate of the underlying probability density function is given by [19]

$$q(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} K_\sigma(\mathbf{x}-\mathbf{x}_i) \tag{2}$$

An important consideration in kernel density estimation is the selection of the kernel size parameter. While there are many techniques for optimizing this parameter such as those based on the maximum likelihood principle [20], a convenient choice is Silverman's rule-of-thumb, which is optimal if the true underlying data distribution is Gaussian. For $n$-dimensional $N$-sample dataset, denoting the sample covariance estimate by $\mathbf{\Sigma_x}$, this gives [15]

$$\sigma^2 = (1/n)tr(\mathbf{\Sigma_x})\left(4/((2n+1)N)\right)^{2/(n+4)} \tag{3}$$

Alternative kernel size selections can be utilized, as well as variable size kernel density estimates. We leave the latter to a future study, as introducing individual kernel sizes for each sample will increase the overall computational complexity of the algorithm.

### 2.3. Fixed-Point Iterations for Vector Quantization

The modes of the data distribution provide a natural clustering solution, where the attraction basin of each mode is a cluster associated with the corresponding mode.[1] Some versions of the mean-shift clustering algorithm rely on this natural clustering definition to determine the clustering solution [17,18]. Given the kernel density estimate of (2), it is easy to determine a fast fixed-point algorithm that determines to which mode each sample belongs. At the peak of each mode, the gradient becomes zero:

$$\frac{\partial q(\mathbf{x})^T}{\partial \mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N}\frac{\partial K_\sigma(\mathbf{x}-\mathbf{x}_i)}{\partial \mathbf{x}} = \mathbf{0} \tag{4}$$

Specifically for a circular Gaussian kernel this becomes

$$\frac{\partial q(\mathbf{x})^T}{\partial \mathbf{x}} = -\frac{1}{N\sigma^2}\sum_{i=1}^{N} G_\sigma(\mathbf{x}-\mathbf{x}_i)\ (\mathbf{x}-\mathbf{x}_i) = \mathbf{0} \tag{5}$$

Isolating $\mathbf{x}$ on one side and reorganizing the terms in (5), we obtain the following fixed-point recursive update for finding the mode corresponding to an arbitrary initial point.

$$\mathbf{x} \leftarrow \left(\sum_{i=1}^{N} G_\sigma(\mathbf{x}-\mathbf{x}_i)\mathbf{x}_i\right)\Bigg/\left(\sum_{i=1}^{N} G_\sigma(\mathbf{x}-\mathbf{x}_i)\right) \tag{6}$$

In general, however, one cannot expect each mode to be a *meaningful* cluster due to the existence of statistical variations in nonparametric density estimation in the finite sample case. Each mode at best represents a vector quantization solution that must be evaluated for the final clustering label assignments appropriately to take into account such effects. The method to resolve this issue will be detailed in the next section.

*Outline of the overall algorithm*

1. Get the data $\mathbf{x}$ and employ the fixed-point algorithm in (6) to find the modes of the probability density function.
2. Construct $\mathbf{K}$, using (9) calculate $D_{ij}$ for all $i,j$ and using (8) construct $\tilde{\mathbf{K}}$.
3. Sort all pairwise affinities defined in non-diagonal entries of $\tilde{\mathbf{K}}$ in an ascending order. The diagonal entries can be ignored, since they all are equal to unity. Representing the affinities of modes with themselves, these entries don't carry out information.
4. Remove the weakest connection, defined by the smallest affinity.
5. Check graph connectivity, and determine the number of separate trees. If the number of separated trees in the graph is equal to the required number of clusters, assign the connected modes into the same cluster and stop. Otherwise go to step 4.

### 2.4. Quantization of the Affinity Matrix

In spectral clustering, the data affinity matrix is constructed by evaluating all pairwise $K_\sigma(\mathbf{x}_j - \mathbf{x}_i)$ similarity measures between samples leading into

$$\mathbf{K}_{ij} = K_\sigma(\mathbf{x}_j - \mathbf{x}_i) \tag{7}$$

Since reducing the size of the affinity matrix is central to the proposed approach, choosing how to quantize the affinity matrix is the most important step in the algorithm. Considering the modes determined by the fixed-point algorithm as an intermediate clustering step, representing each mode with a single entry is a suitable way of quantizing the affinity matrix. Several methods can be applied to define this entry that will represent the whole set of data points that are assigned to it; since it is known to be a reliable divergence measure, the normalized graph cut merits special attention at this point. Defining a symmetric affinity metric between clusters of points, the normalized graph cut leads to a positive semi definite quantized affinity matrix $\tilde{\mathbf{K}}$ defined as

$$\tilde{\mathbf{K}}_{ij} = \frac{D_{ij}}{\sqrt{D_{ii}}\sqrt{D_{jj}}} \tag{8}$$

where $D_{ij}$ is the normalized graph cut between intermediate clusters, namely modes, $i$ and $j$ is defined as,

$$D_{ij} = \sum_{k\in i}\sum_{l\in j} K(\mathbf{x}_k^i - \mathbf{x}_l^j) = \sum_{k\in i}\sum_{l\in j}\mathbf{K}_{kl} \tag{9}$$

Investigating (8) and (9), one can also interpret $\tilde{\mathbf{K}}_{ij}$ as a probability density estimate distance between modes $i$ and $j$, since normalized graph cut is an inner product between the probability density functions of individual

[1] The attraction basin of a mode is the volume in the data space from which gradient ascent on the data probability density function leads to the peak of that mode.

modes, whose result is equal to the cosine of the angle between the means of individual modes in the kernel induced feature space. Specifically, one can easily see that if

$$D_{ij} = \int p_i(\mathbf{x}) p_j(\mathbf{x}) d\mathbf{x} \quad \forall i, j \tag{10}$$

where $p_i(\mathbf{x})$ and $p_j(\mathbf{x})$ are the probability density functions of the individual clusters. Then (9) is a Parzen window based estimate for this inner product. Furthermore, $\tilde{\mathbf{K}}_{ij}$ in (8) is the CS-distance between $p_i(\mathbf{x})$ and $p_j(\mathbf{x})$ [20].

### 2.5. Spectral Clustering of Quantized Affinity Matrix

Spectral clustering methods employ the eigendecomposition of the affinity matrix to to obtain a clustering statistic. The difficulty, however, is that the data affinity matrix is $N \times N$, for an $N$-sample dataset, and eigenvector calculations in high dimensionality are computationally very expensive – $O(N^3)$.

After evaluating the quantized affinity matrix $\tilde{\mathbf{K}}$, one can also use any well-known spectral clustering method in the literature. However, we propose another simple but robust algorithm here, which would become impractical for large affinity matrices on the orders of data sizes due to its $O(N^4)$ complexity. On the other hand, this method produced good results for small sized quantized affinity matrices and preferred here to be able to indicate that quantizating the affinity matrix in a suitable way, one can use a variety of different spectral clustering algorithms using the resulting quantized matrix, which were originally impractical to use for huge data affinity matrices. Once the clustering results for the modes are obtained, the actual clustering can be achieved by assigning a common label to all the data points in the same cluster of modes.

The spectral clustering algorithm used here is as simple as sorting all affinities in $\tilde{\mathbf{K}}$ with an ascending order and removing the weakest connection defined by the smallest affinity one by one until the required number of clusters is reached. In each step, the graph connectivity is being checked and the algorithm decides on either continuing to remove connections or stopping and assigning the connected modes into the same cluster. Performed in each iteration with $O(N^2)$ complexity, checking the graph connectivity is the dominant computational load, resulting in a $O(N^4)$ complexity for the overall algorithm. To check the graph connectivity, a well-known connected components algorithm is used [21]. The outline of the resulting spectral clustering algorithm is given in Table 1.

## 3. EXPERIMENTAL RESULTS

*Crescent Dataset*: This dataset consists of two crescent-shaped clusters with a nonlinear separation boundary in between. For each cluster, 200 two-dimensional samples are generated by uniformly selecting the angle in a $\pi$-radian arc and perturbing the radius with Gaussian distributed random values. The class centers are selected such that the possibility of having a linear boundary on which the classes become easily separable is eliminated.

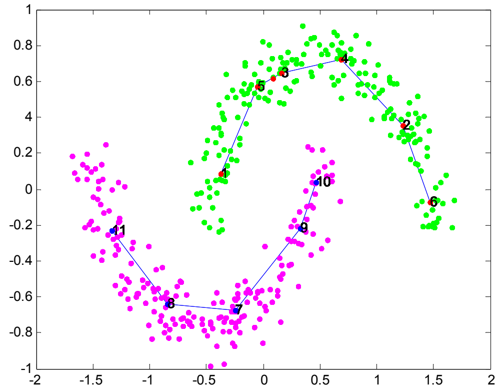

Figure 2. The original crescent dataset, where the points represent the data points and the lines connect the modes that are clustered into the same cluster.

The original dataset and a sample simulation result are presented in figure 2, where the points represent data and the connected intermediate clusters represent the clustering result. Original affinity matrix $\mathbf{K}$ and quantized affinity matrix $\tilde{\mathbf{K}}$ are shown figure 3a and figure 3b.

The results in figure 2 demonstrate a *perfect* clustering performance for this dataset (since clusters are reasonably separated). Additionally, for this example one can also notice the similarities between $\mathbf{K}$ and $\tilde{\mathbf{K}}$, which is not the case in general. Our experiments with various degrees of cluster overlap yielded consistent and reasonable clustering solutions.

Since this dataset is synthetically generated and the clusters are designed such that their distributions show similarities rather than a translation and rotation in the original space, not surprisingly, the number of modes in two clusters turned out to be almost equal. Generally, since the convergence rate in the fixed-point iterations is constant throughout the data feature space, clusters with different in-cluster-variances in the same dataset may result in different numbers of modes, hence different number of intermediate clusters, for each individual cluster. Although synthetically generated, this dataset is quite successful in showing the basic concepts applied, and result obtained with a real dataset will be presented in the next subsection.

*Handwritten Digit Recognition*: Originally being generated for a classification problem, clustering handwritten digit data is a suitable real-data application for demonstrating the proposed clustering algorithm. This digit database contains 250 samples from 44 subjects and can be found in UCI database [22].

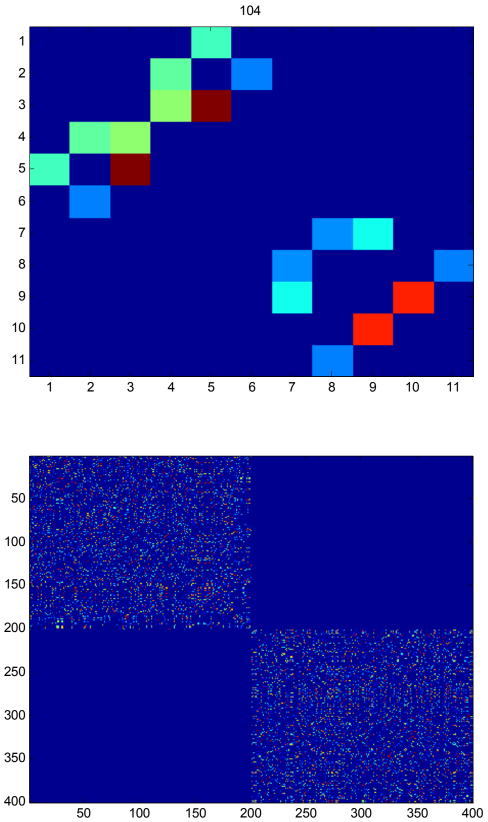

Figure 3. Quantized (top) and original (bottom) affinity matrices for the crescent dataset (diagonals nulled).

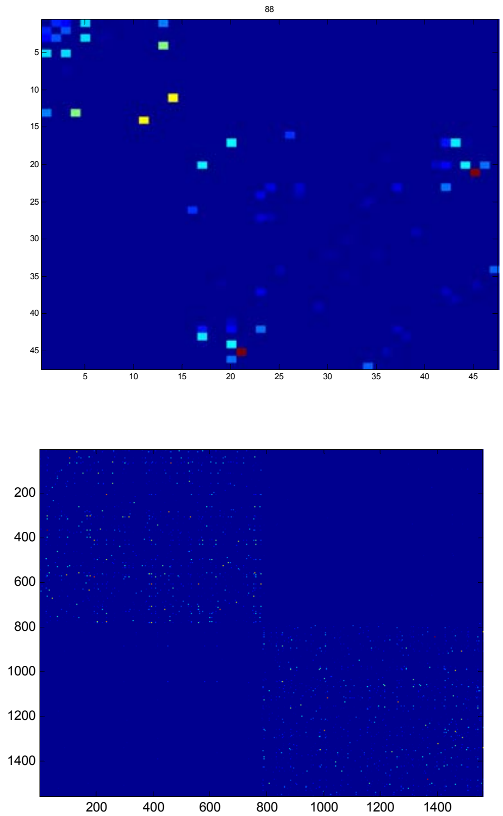

Figure 4. Quantized (top) and original (bottom) affinity matrices for handwritten digits (diagonals nulled).

Although the original database contains ten digits, for ease of illustration and discussion, we utilize only the digits 1 and 2.

Being sixteen-dimensional, the original data is impossible to present in a figure even with a suitable two-dimensional subspace projection. For this dataset, the quantized affinity matrix $\tilde{\mathbf{K}}$ is presented along with the original affinity matrix $\mathbf{K}$ in Figure 3a and Figure 3b, respectively. Comparing $\mathbf{K}$ and $\tilde{\mathbf{K}}$ one can easily observe the effect of different in-cluster-variances in the results of this dataset; namely, the number of intermediate clusters in cluster representing one is less than those in cluster representing two.

Although presented here for only two class problems, the method can be easily extended into multi-cluster problems by adjusting the required number of clusters at output to the desired. Automatic selection of the number of clusters based on the affinity matrices is also possible and will be the subject of future work.

## 4. CONCLUSIONS

Although proven to be effective and considered to be the state-of-the-art methods for clustering, the main drawback of spectral clustering methods is the computational burden. This is mainly caused by the calculation of the eigenvalues of the affinity matrix, having a $O(N^3)$ complexity. In this paper, a mean shift preprocessing stage is proposed along with a spectral clustering algorithm that simply uses pairwise similarities.

This technique may lead to a sub-optimal solution for an unsuitably selected kernel function, due to a poor estimate of data probability density. On the other hand, the absence of theory for a suitable selection for the kernel is a common drawback of all spectral clustering algorithm. The mean shift preprocessing stage proved to be practical, providing useful results with a significant decrease in overall computational complexity.

Future work will focus on the selection of optimal kernels, variable-size kernel density estimation for better results in the mean-shift stage, and automatic detection of the number of *statistically significant* clusters.

## REFERENCES

[1] M. Fiedler, “Algebraic Connectivity in Graphs,” Czechoslovak Mathematics Journal, vol. 23, pp. 298-305, 1973.

[2] S. Sarkar, P. Soundararajan, “Supervised Learning of Large Perceptual Organization: Graph Spectral Partitioning and Learning Automata,” IEEE

Transactions on Pattern Analysis and Machine Intelligence,vol. 22, no. 52, pp. 504-525, 2000.

[3] A.Y. Ng, M. Jordan, Y. Weiss, “On Spectral Clustering: Analysis and an Algorithm,” Advances in Neural Information Processing Systems, vol. 14, no. 2, pp. 849-856, 2001.

[4] R. Kannan, S. Vempala, A. Vetta, “On Clusterings: Good, Bad and Spectral,” EEE Foundations of Computer Science, pages 367-377, Redondo Beach, CA, USA, 2000.

[5] J. Shi, J. Malik, “Normalized Cuts and Image Segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no.8, pp. 888-905, 2000.

[6] M. Meila, L. Xu, “Multiway Cuts and Spectral Clustering,” Technical Report 442, University of Washington, Department of Statistics, January 2004.

[7] P. Perona, W. T. Freeman, “A Factorization Approach to Grouping,” Proc. European Conference on Computer Vision, pp. 655-670, 1998.

[8] Y. Weiss, “Segmentation Using Eigenvectors: A Unifying View,” International Conference on Computer Vision, pages 975-982, 1999.

[9] C. Alpert, S. Yao, “Spectral Partitioning: The More Eigenvectors the Better,” ACM/IEEE Design Automation Conference, 1995.

[10] Y. Azar, A. Fiat, A. Karlin, F. McSherry, J. Saia, “Spectral Analysis of Data,” 33rd Symposium on Theory of Computing, pp. 619-626, 2001.

[11] P. Chang, D. Schlag, J. Zien, “Spectral K-Way Ratio-Cut Partitioning and Clustering,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 13, no. 9, pp. 1088-1096, 1994.

[12] G. Scott, H. Longuet-Higgins, “Feature Grouping by Relocalisation of Eigenvectors of the Proximity Matrix,” British Machine Vision Conference, pp. 103-108, 1990.

[13] D. J. Higham, M. Kibble, “A Unified View of Spectral Clustering,” Technical Report 02, University of Strathclyde, Department of Mathematics, January 2004.

[14] R. Jenssen, T. Eltoft, J. C. Principe, “Information Theoretic Spectral Clustering,” In International Joint Conference on Neural Networks, pp. 111-116, Budapest, Hungary, 2004a.

[15] B.W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London, 1986.

[16] R. Jenssen, D. Erdogmus, J. C. Principe, T. Eltoft, “The Laplacian PDF Distance: A Cost Function for Clustering in a Kernel Feature Space,” Accepted to 18'th Annual Conference on Neural Information Processing Systems, Vancouver, B.C., Canada, 2004b.

[17] D. Comaniciu, P. Meer, “Mean Shift: A Robust Approach Toward Feature Space Analysis,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619, 2002.

[18] B. Georgescu, I. Shimshoni, P. Meer, “Mean Shift Based Clustering in High Dimensions: A Texture Classification Example,” Proceedings of ICCV’03, pp. 456-463, 2003.

[19] E. Parzen, “On the Estimation of a Probability Density Function and the Mode,” Ann. Math. Stat., vol. 32, pp. 1065-1076, 1962.

[20] L. Devroye, G. Lugosi, *Combinatorial Methods in Density Estimation*, Springer, New York, 2001.

[21] Thomas H. Cormen, Charles E. Leiserson, Roland L. Rivest, *Introduction to Algorithms*, MIT Press and McGraw-Hill, New York, 1990.

[22] UCI Machine Learning Repository, http://www.ics.uci.edu/~mlearn/MLSummary.html