Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 27 (2006) 1650–1664
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Image segmentation by unsupervised sparse clustering ☆

Byoung-Ki Jeon [a], Yun-Beom Jung [b], Ki-Sang Hong [b,*]

[a] Samsung Electronics Co., Ltd., Dong Suwon P.O. Box 105, 416, Maetan-3Dong, Yeongtong-Gu, Suwon 442-600, Republic of Korea
[b] Electrical and Computer Engineering Division, POSTECH, San 31 Hyoja-Dong, Pohang 790-784, Republic of Korea

Received 12 August 2004; received in revised form 16 January 2006
Available online 27 June 2006

Communicated by Y.J. Zhang

Abstract

In this paper, we present a novel solution for image segmentation based on positiveness which regards the segmentation as a graph-theoretic clustering problem. Contrary to spectral clustering methods using eigenvectors, the proposed method tries to find an additive combination of positive components from an originally positive data-driven matrix. By using the positiveness constraint, we obtain sparsely clustered results which do not include cancellations by negative entries. Thus, we call this method sparse clustering. The proposed method adopts a binary tree structure and solves a model selection problem by automatically determining the number of clusters using intra- and inter-cluster measures. We tested our method with image sequences as well as single frame data such as points and gray-scale, color, and texture images. Moreover, in order to objectively evaluate the performance of our method, we compared the results of the proposed method with those of the human segmentation and the Ncut method using various images including the Berkeley datasets. Experimental results show that the proposed method provides very successful and encouraging segmentations.

Keywords: Segmentation; Positiveness; Sparse clustering; Binary tree; Model selection; Intra- and inter-cluster measures

1. Introduction

Image segmentation is used to distinguish objects from their background or to partition an image into related regions. It can be one of the primary steps in image analysis and object recognition. It is also one of the most popular problems in computer vision and is closely related to perceptual grouping, which is based on human psychology and has a long history (Wertheimer, 1938). Since the performance evaluation of image segmentation is largely subjective, there is unfortunately no standard or perfect segmentation algorithm. Therefore, most previous works tried to solve the image segmentation problem by first establishing the criterion to be optimized and next by developing an algorithm to optimize it.

Many researchers have approached image segmentation from several perspectives: point-based approaches by thresholding (Ohlander, 1975; Cheriet et al., 1998), contour-based approaches by edge detection (Perkins, 1980; Prager, 1980), region-based approaches by region merge and split (Adams and Bischof, 1994; Chang and Li, 1994), clustering-based approaches (Lucchese and Mitra, 1999; Shi and Malik, 2000), and other optimization-based approaches using a Bayesian framework, neural networks (Geman and Geman, 1984; Campbell et al., 1997). The clustering approaches can be categorized into two methods: *density-based clustering* methods, such as *k*-means and EM clustering (Lucchese and Mitra, 1999; Moore, 1999), and *graph-theoretic clustering* methods using data-driven matrices (Shi and Malik, 2000; Fiedler, 1975; Hagen and Kahng,

☆ Part of this paper was published as a conference paper in WACV2005 (Jeon et al., 2005).

* Corresponding author. Tel.: +82 54 279 2216; fax: +82 54 279 5586.
E-mail addresses: byoungki.jeon@samsung.com (B.-K. Jeon), jyb@postech.ac.kr (Y.-B. Jung), hongks@postech.ac.kr (K.-S. Hong).

0167-8655/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2006.03.011

1992; Ding et al., 2001). The proposed method in this study belongs to the latter and is compared with *spectral clustering*, the representative graph-theoretic clustering method.

In a graph-theoretic framework, we consider that original data, such as pixel values in a given image, constitute a weighted undirected graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$. Each node in $\boldsymbol{V}$ is a feature vector calculated from each original datum, and an edge in $\boldsymbol{E}$ is linked between every pair of nodes. The edge weight $w(i,j)$ is a function of the distance between nodes i and j in feature space and it is mainly defined as to be larger as the distance gets smaller. If a matrix is constructed where each element is the edge weight between a pair of nodes, then it will be a square symmetric matrix representing adjacencies between graph nodes and having the same number of rows (or columns) as the number of nodes in $\boldsymbol{V}$. In spectral clustering, the objective function to partition the graph $\boldsymbol{G}$ is first defined and next optimized using an efficient algorithm. Since its solution eventually leads to the eigenvectors of the *graph adjacency matrix* – here, we call it the *affinity matrix* – we can define the spectral clustering methods as algorithms that cluster points using eigenvectors of the data-driven matrix. Spectral clustering has its origin in spectral graph partitioning (Fiedler, 1975), and some algorithms have successfully optimized criteria involved in graph cuts such as ratio-cut (Hagen and Kahng, 1992), normalized cut (Shi and Malik, 2000), and min–max cut (Ding et al., 2001).

According to the *spectral theorem* (also called *principal axis theorem*) which is the theoretical basis of spectral clustering, a real symmetric matrix, such as the affinity matrix, can be factored into $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^{\mathrm{T}} = \lambda_1 \boldsymbol{q}_1 \boldsymbol{q}_1^{\mathrm{T}} + \lambda_2 \boldsymbol{q}_2 \boldsymbol{q}_2^{\mathrm{T}} + \cdots + \lambda_n \boldsymbol{q}_n \boldsymbol{q}_n^{\mathrm{T}}$ with the orthonormal eigenvectors $\boldsymbol{q}_i$'s in $\boldsymbol{Q}$ and the eigenvalues λ_i's in $\boldsymbol{\Lambda}$ (Strang, 1988). Since the affinity matrix actually represents *similarities* between nodes, it cannot have any negative entries. For a matrix with only positive elements, we can observe that its largest eigenvector has only positive entries but the subsequent eigenvectors can have negative entries because of the orthogonality constraints between eigenvectors. In other words, spectral clustering can lead to a subtractive combination of negative components from the originally positive affinity matrix and this can cause complex cancellations which may prevent us from obtaining intuitive meanings from the factorized components.

Several previous studies have been proposed for factorization using positiveness or non-negativeness constraints (Paatero and Tapper, 1994; Lee and Seung, 1999; Welling and Weber, 2001). There are two main reasons why positiveness is of interest. The first one is that positiveness makes intuitive interpretation possible. If we apply one of the spectral clustering methods to originally positive data such as pixel values or the affinity matrix, an interpretation of the eigenvectors with negative entries is often impossible. Further, these negative entries can cause a subtractive combination with complex cancellations. The second reason is that the positiveness property could be a key to the ways that the human brain perceives objects – there is psychological and physiological evidence for parts-based representations in the brain (Lee and Seung, 1999). Since positiveness allows only additive, not subtractive, combinations, it can lead to parts-based representations and eventually provide *sparse codings* where many entries of each factor are zero (Olshausen and Field, 1996; Chennubhotla and Jepson, 2001). The sparse coding represents an observation vector using only few sparse basis vectors and is different from *distributed coding*, such as *principal component analysis (PCA)*, which represents an observation using most of the basis vectors, which are holistic.

By incorporating the preceding properties into the clustering framework, we propose a novel clustering method which factorizes the originally positive affinity matrix under the positiveness constraint using the *positive tensor factorization (PTF)* method (Welling and Weber, 2001). The proposed method provides sparsely clustered results based on positiveness, so we call it *sparse clustering*.

Since the image segmentation problem can be regarded as inherently hierarchical – human beings first segment an image into several large regions and then successively segment those regions in details, the proposed method adopts a *hierarchical structure* based on the *binary tree*. In this structure, we use both *intra- and inter-cluster measures* to determine whether the current cluster should be factorized or not, and solve the *model selection* problem by automatically determining the number of clusters based on these two measures. If we only use a single measure, then the hierarchical structure may cause over- or under-segmentation.

This paper is organized as follows. In Section 2, we explain how the affinity matrix is factorized under the positiveness constraint using PTF. In Section 3, the overall procedure of the proposed method is provided, including the binary tree structure and intra- and inter-cluster measures. Section 4 presents the experimental results and their analysis for image sequences as well as static images including some from the Berkeley datasets.[1] To objectively analyze the segmentation results, we compare the performance of our method with that of the human segmentation as well as the *normalized cut (Ncut)* method (Shi and Malik, 2000), which is the representative spectral clustering method for image and motion segmentation. Finally, conclusions and future works are discussed in Section 5. Some parts of this work were presented as a conference paper (Jeon et al., 2005).

2. Positiveness-based factorization

As mentioned in Section 1, the positiveness-based factorization of originally positive data has encouraging characteristics – the intuitive interpretation and the additive combination of positive and sparse components without

[1] The Berkeley datasets are available at http://www.cs.berkeleyedu/projects/vision/grouping/segbench/.

complex cancellations. Due to this, much research has been done on factorization such as positive matrix factorization (PMF) (Paatero and Tapper, 1994), non-negative matrix factorization (NMF) (Lee and Seung, 1999), and positive tensor factorization (PTF) (Welling and Weber, 2001). We selected PTF because of its generality. These approaches have iterative learning rules which are not additive but multiplicative and therefore can maintain positiveness by initializing components as positive and multiplying positive scale factors during iterations. Before explaining the factorization method in detail, we need to define a matrix which represents similarity information between data.

2.1. Features and the affinity matrix

In this paper, we try to solve the image segmentation problem in a graph-theoretic framework where the given image data constitutes a weighted undirected graph $\boldsymbol{G}=(\boldsymbol{V},\boldsymbol{E})$. To segment an image means to partition the graph $\boldsymbol{G}$, which is performed by factorizing an affinity matrix which measures similarities between all of the graph nodes in $\boldsymbol{V}$. Therefore, the affinity matrix is a square symmetric matrix with the same number of rows (and columns) as the number of nodes and is defined using not only feature similarities but also spatial adjacencies as follows:

$$A_{ij}=\mathrm{e}^{-\frac{\|\boldsymbol{F}(\boldsymbol{X}_i)-\boldsymbol{F}(\boldsymbol{X}_j)\|^2}{\sigma_{\mathrm{F}}^2}}\times\begin{cases}\mathrm{e}^{-\frac{\|\boldsymbol{X}_i-\boldsymbol{X}_j\|^2}{\sigma_X^2}} & \text{if } \|\boldsymbol{X}_i-\boldsymbol{X}_j\|<r,\\ 0 & \text{otherwise,}\end{cases}\tag{1}$$

where $\boldsymbol{X}_i$ is the spatial position vector of node i, and $\boldsymbol{F}(\boldsymbol{X}_i)$, the feature vector at $\boldsymbol{X}_i$, has different types according to the input data:

- $\boldsymbol{F}(\boldsymbol{X}_i)=1$ for point set data. (For this data, we use the spatial adjacency information only.)
- $\boldsymbol{F}(\boldsymbol{X}_i)=I(\boldsymbol{X}_i)$ for gray-scale images. ($I(\boldsymbol{X}_i)$ is intensity value at $\boldsymbol{X}_i$.)
- $\boldsymbol{F}(\boldsymbol{X}_i)=[r(\boldsymbol{X}_i),g(\boldsymbol{X}_i),b(\boldsymbol{X}_i)]$ for color images. ($r(\boldsymbol{X}_i)$, $g(\boldsymbol{X}_i)$, and $b(\boldsymbol{X}_i)$ are red, green, and blue values at $\boldsymbol{X}_i$, respectively.)
- $\boldsymbol{F}(\boldsymbol{X}_i)=[|I(\boldsymbol{X}_i)*f_1|^2,|I(\boldsymbol{X}_i)*f_2|^2,\ldots,|I(\boldsymbol{X}_i)*f_n|^2]$ for texture images. (The f_i's are *Gabor filters* with different scales and orientations (Jain and Farrokhnia, 1991). The Gabor filters are complex, so we use the squared magnitude responses involved in the *spectrum*.)

To compute the affinity matrix we need to select parameters such as σ_{F} for feature similarities and the neighborhood size r and σ_x for spatial adjacencies. Since these parameters determine the shape of the affinity matrix, their selection is very important; detailed explanations will be given in Section 4.

2.2. PTF algorithm

The PTF considers a tensor of order d, $\boldsymbol{T}_{i_1,\ldots,i_d}$ with positive entries. A tensor of order d in m-dimensional space is a mathematical object that has d indices and m^d components and obeys certain transformation rules. Tensors are generalizations of scalars (order 0), vectors (order 1), and matrices (order 2) to arbitrary higher order arrays (Lovelock and Rund, 1989). Because of this generality, we adopt PTF instead of PMF or NMF. In PTF, the tensor with positive entries is factorized into N positive components, $\boldsymbol{C}_{ij,n}^{(j)}$, as follows:

$$\boldsymbol{T}_{i_1,\ldots,i_d}=\sum_{n=1}^{N}\boldsymbol{C}_{i_1,n}^{(1)}\boldsymbol{C}_{i_2,n}^{(2)}\cdots\boldsymbol{C}_{i_d,n}^{(d)},\tag{2}$$

such that the *reconstruction error*,

$$\mathrm{RE}=\sum_{i_1,\ldots,i_d}\left(\boldsymbol{T}_{i_1,\ldots,i_d}-\sum_{n=1}^{N}\boldsymbol{C}_{i_1,n}^{(1)}\boldsymbol{C}_{i_2,n}^{(2)}\cdots\boldsymbol{C}_{i_d,n}^{(d)}\right)^2\tag{3}$$

is minimized.

The update rule is obtained by taking derivatives of Eq. (3) with respect to each $\boldsymbol{C}_{ij,n}^{(j)}$, finding the point where the derivative vanishes, and defining a positive scaling factor based on the information around that point. The resulting update rule makes the component, $\boldsymbol{C}_{ij,n}^{(j)}$ move in the direction of a negative gradient of the reconstruction error. For the arbitrary index i_j, we define $\boldsymbol{B}_{ij,K}=\boldsymbol{T}_{i_1,\ldots,i_d}$, grouping $i_1,\ldots,i_{j-1},i_{j+1},\ldots,i_d$ into one index K, and $\boldsymbol{D}_{K,n}=\boldsymbol{C}_{i_1,n}^{(1)}\cdots\boldsymbol{C}_{i_{j-1},n}^{(j-1)}\boldsymbol{C}_{i_{j+1},n}^{(j+1)}\cdots\boldsymbol{C}_{i_d,n}^{(d)}$. While leaving the other components fixed, we update the component $\boldsymbol{C}_{ij,n}^{(j)}$ using the rule,

$$\boldsymbol{C}^{(j)}\leftarrow\boldsymbol{C}^{(j)}\times\frac{\boldsymbol{B}\cdot\boldsymbol{D}}{\boldsymbol{C}^{(j)}\cdot\boldsymbol{D}^{\mathrm{T}}\cdot\boldsymbol{D}},\tag{4}$$

where '$\times$' and the fraction mark '—' denote component-wise multiplication and division, and '$\cdot$' denotes matrix multiplication. By initializing all $\boldsymbol{C}_{ij,n}^{(j)}$ with random positive values and scaling them with a positive scale factor, we keep the values of $\boldsymbol{C}_{ij,n}^{(j)}$ positive.

2.3. Factorization of the affinity matrix

In this section, we explain how we can factorize the affinity matrix using the PTF algorithm from the preceding section, and how we can extract segmentation information from the factorized components. The affinity matrix $\boldsymbol{A}$ can be regarded as a tensor of order 2 ($d=2$), which should be factorized into two positive components ($N=2$) since the proposed method adopts the hierarchical binary tree structure to recursively bi-segment images. If y and x are indices denoting rows and columns of the given image, an element of the affinity matrix $\boldsymbol{A}$ can be expressed according to the tensor notation of Eq. (2) as follows:

$$\boldsymbol{A}_{(y,x),(y,x)}=\boldsymbol{A}_{I,J}=\boldsymbol{A}_{J,I}=\sum_{n=1}^{2}\boldsymbol{C}_{I,n}^{(1)}\boldsymbol{C}_{J,n}^{(2)}\tag{5}$$

$$=\boldsymbol{C}_{I,1}^{(1)}\boldsymbol{C}_{J,1}^{(2)}+\boldsymbol{C}_{I,2}^{(1)}\boldsymbol{C}_{J,2}^{(2)}$$

$$=\boldsymbol{C}_{I,1}^{(1)^2}+\boldsymbol{C}_{I,2}^{(1)^2}.\tag{6}$$

Since the affinity matrix is symmetric, the indices denoting rows and columns can be exchanged with each other: $\boldsymbol{A}_{I,J} = \boldsymbol{A}_{J,I}$. Therefore, the component of $\boldsymbol{C}^{(1)}_{I,n}$ is equal to $\boldsymbol{C}^{(1)}_{J,n}$, which is again equal to $\boldsymbol{C}^{(2)}_{J,n}$ since the superscripts such as (1) and (2) just represent the order of indices. Eq. (6) is obtained by the equalities $\boldsymbol{C}^{(1)}_{I,n} = \boldsymbol{C}^{(1)}_{J,n} = \boldsymbol{C}^{(2)}_{J,n}$, so we need to update only one matrix $\boldsymbol{C}^{(1)}$ with two columns where each element of each column represents one of two positive components for our bi-partitioning. By manipulating matrices such as $\boldsymbol{B}$ and $\boldsymbol{D}$ using the derived equalities and substituting them in Eq. (4), we obtain a simplified update rule of the matrix $\boldsymbol{C}^{(1)}$ for the index $i_j = (y, x) = I$ as follows:

$$\boldsymbol{B}_{I,J} = \boldsymbol{A}_{I,J}, \quad \boldsymbol{D}_{J,n} = \boldsymbol{C}^{(2)}_{J,n} = \boldsymbol{C}^{(1)}_{I,n}, \tag{7}$$

$$\boldsymbol{C}^{(1)}_{[P\times N]} \leftarrow \boldsymbol{C}^{(1)}_{[P\times N]} \times \left(\frac{\boldsymbol{A}_{[P\times P]} \cdot \boldsymbol{C}^{(1)}_{[P\times N]}}{\boldsymbol{C}^{(1)}_{[P\times N]} \cdot \boldsymbol{C}^{(1)^{\mathrm{T}}}_{[N\times P]} \cdot \boldsymbol{C}^{(1)}_{[P\times N]}} \right)^{1/2}, \tag{8}$$

where P is the number of pixels (or graph nodes) and N is the number of positive components, and the dimensions of matrices used for update are indicated with subscripts. In Eq. (8), the power of 1/2 is added to reduce the step size and thus reduce the chance of overshooting the minimum. From preceding results of the tensor factorization, we can observe that the affinity matrix is the product of two matrices with positive entries such that $\boldsymbol{A} = \boldsymbol{C}^{(1)} \cdot \boldsymbol{C}^{(2)^{\mathrm{T}}} = \boldsymbol{C}^{(1)} \cdot \boldsymbol{C}^{(1)^{\mathrm{T}}}$. Further, if we factorize other matrices where columns are observation vectors unlike the affinity matrix, this expression is analogous to that of subspace analysis methods such as PCA and ICA (independent component analysis). In those methods, the subspace is represented as a linear combination $\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{S}$ where an observation vector (each column of $\boldsymbol{X}$) is obtained by multiplying basis vectors (all columns of $\boldsymbol{Q}$) by an encoding (the column of $\boldsymbol{S}$ in the same position as the corresponding observation vector).

In our binary tree structure, the matrix $\boldsymbol{C}^{(1)}$ has two columns ($N = 2$) which can be allocated to the left and right nodes of the binary tree. From $\boldsymbol{C}^{(1)}$, we can obtain the segmentation information as follows. $\boldsymbol{C}^{(1)}$ has P rows where each row corresponds to each pixel position and has two non-negative values. After convergence by Eq. (8), one of them is probably a somewhat large positive value and the other a negligible positive value or zero. Therefore, if the value of the first column is larger than or equal to that of the second column, then the corresponding pixel is assigned to the left node, and vice versa – this bi-partitions the graph or bi-segments the image. This procedure is a kind of *discretization* of continuous factorized components we call the *comparison-based method*. The discretization problem is also discussed in the Ncut paper of Shi and Malik (2000), which bi-partitions the graph by selecting the thresholding point such that the resulting partition of the eigenvector has the best normalized cut value. Like this, we can bi-partition the cluster by thresholding a factorized component which corresponds to one of two columns of $\boldsymbol{C}^{(1)}$, say, the first column. The threshold value is determined by finding the splitting point such that the resulting partition has the smallest inter-cluster score; this is called the *thresholding-based method*. In our implementation, the two proposed methods show very similar discretized results. However, the comparison-based method may provide unstable segmentation boundaries if the compared values do not differ by a large enough margin. Therefore, for the experimental results in Section 4, we adopt the thresholding-based discretization method.

3. Unsupervised sparse clustering

In this section, we explain the main procedure and structure of the proposed method, including intra- and inter-cluster measures. Since our method clusters points based on PTF which provides sparse codings and determines the number of clusters based on the hierarchical structure without user interaction, we call it *unsupervised sparse clustering*.

3.1. Overall flow and hierarchical structure

Fig. 1 shows the overall flow of the proposed method. We first compute feature vectors according to the input data types as explained in Section 2.1. Next, we compute the affinity matrix from them and factorize it into two positive components using PTF. Since we use the binary tree, which is one hierarchical structure, we need to bi-factorize the affinity matrix. For each untested leaf node that is one of the factorized components, we compute the intra-cluster measure h_1, and, if it is larger than or equal to a threshold

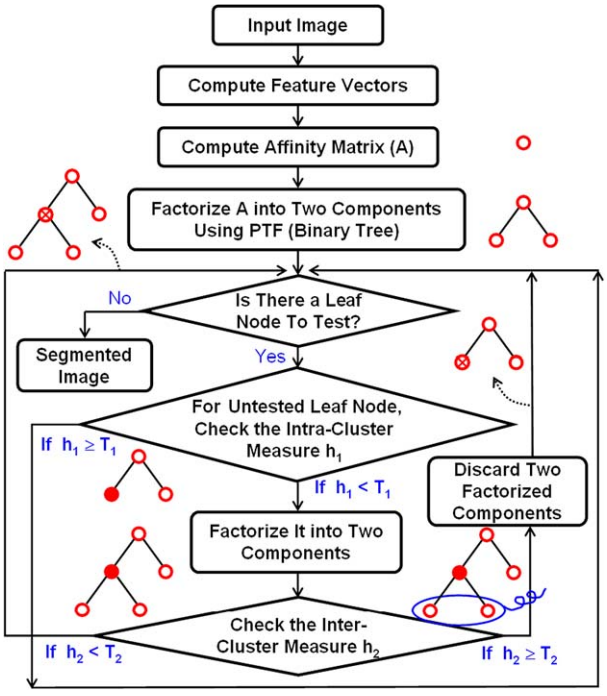

Fig. 1. Overall flow of unsupervised sparse clustering.

T_1, then we go to the next untested leaf node. If it is smaller than T_1, we factorize that node into two positive components, and compute the inter-cluster measure h_2 between them. If it is larger than or equal to another threshold T_2, then we remove these factorized components and go to the next leaf node, and if not, we accept the components as new left and right leaf nodes and also go to the next leaf node. This procedure is iterated until there is no leaf node to test, eventually leading to hierarchically segmented results.

3.2. Intra- and inter-cluster measures

The proposed method automatically determines the number of clusters by incorporating two measures into the binary tree structure. The first one is the *intra-cluster measure* which determines if the current cluster should be factorized or not, and the second one is the *inter-cluster measure* which determines if the factorization by the intra-cluster measure is acceptable or not by checking separability between the two factorized components.

Spectral graph theory provides a measure of tightness of clusters called the *Cheeger constant* (Chung, 1997). If there is a partition of $(\mathscr{I}, \overline{\mathscr{I}})$ for the current cluster C_i, the Cheeger constant $\phi(C_i)$ is defined as

$$\phi(C_i) = \min_{\mathscr{I}} \frac{\sum_{j\in\mathscr{I},k\in\overline{\mathscr{I}}} A_{jk}^{(i)}}{\min\{\mathrm{vol}(\mathscr{I}), \mathrm{vol}(\overline{\mathscr{I}})\}}, \tag{9}$$

where $A_{jk}^{(i)}$ is an affinity value between nodes j and k in the cluster C_i. The volume of a partition or a subset $\mathscr{I}$ of the cluster C_i is denoted by $\mathrm{vol}(\mathscr{I})$ and defined as $\sum_{x\in\mathscr{I}}\deg_x$. $\deg_x$, which denotes the degree of a graph node x in $\mathscr{I}$, means the number of nodes adjacent to x. If the weight of the edges across the partition is small and each of the partitions has a moderately large volume, then the Cheeger constant is small – this means the cluster C_i is not tight enough and can be factorized more. However, to compute $\phi(C_i)$ is to find the optimal partition of $(\mathscr{I}, \overline{\mathscr{I}})$, which is a *NP-hard* problem. Therefore, we actually use the *eigengap* which is closely related to the Cheeger constant (Kannan et al., 2000; Ng et al., 2001). The eigengap of $\delta(C_i)$ for the cluster C_i is defined as $1 - \lambda_2/\lambda_1$, where λ_1 and λ_2 are the largest and the second largest eigenvalues of the affinity matrix for the pixels in C_i. The eigengap provides bounds of the Cheeger constant by the inequality

$$\frac{\phi^2}{2} \leqslant \delta \leqslant 2\phi, \tag{10}$$

and therefore we define the intra-cluster measure of the cluster C_i as the eigengap which is related to the lower-bound of the Cheeger constant ($\phi \geqslant \delta/2$) as follows:

$$\mathrm{Intra}(C_i) = h_1(C_i) = \delta(C_i). \tag{11}$$

If the intra-cluster measure h_1 is smaller than the threshold T_1, then we regard the cluster as not tight enough and factorize it more.

In cases where we only consider intra-cluster information, the proposed method may provide over- or under-segmented results – if a large threshold T_1 is used to factorize one cluster, other clusters may be erroneously factorized, thus over-estimating the number of clusters. Of course, the opposite scenario is also possible. To prevent this, we adopt another measure, called the inter-cluster measure h_2, to check the separability between the two sub-clusters previously factorized by the intra-cluster measure. It is defined as

$$\mathrm{Inter}(C_j, C_k) = h_2(C_j, C_k) = \sum_{j\in C_j}\sum_{k\in C_k}\frac{A_{jk}^2}{d_j d_k}, \tag{12}$$

where d_j denotes $\sum_{j,n\in C_j} A_{jn}$ and measures how strongly the node $j \in C_j$ is connected to the rest of C_j, and $\sum_{k\in C_k} A_{jk}$ measures how strongly the node j is connected to the nodes in the other cluster (Ng et al., 2001). If the inter-cluster measure h_2 is smaller than the threshold T_2, then all nodes are connected more to nodes in the same cluster than to nodes in the other cluster, and thus the factorization by the intra-cluster measure is acceptable – if not, the two factorized sub-clusters are unacceptable and should be removed.

The proposed inter-cluster measure has a similar form as the *min–max cut*, which can be represented according to our notations as follows (Ding et al., 2001):

$$\begin{aligned}\text{min}-\text{max cut} = &\frac{\sum_{j\in C_j}\sum_{k\in C_k} A_{jk}}{\sum_{j\in C_j} d_j} \\ &+ \frac{\sum_{j\in C_j}\sum_{k\in C_k} A_{jk}}{\sum_{j\in C_k} d_k}.\end{aligned} \tag{13}$$

If we first replace the addition with a multiplication ignoring summation marks ($\sum' s$) and next sum up this quantity with respect to all nodes in the clusters C_j and C_k, then this criterion is exactly the same as our inter-cluster measure in Eq. (12). Since both criteria are based on the same quantities such as d_j, d_k, and A_{jk}, despite slightly different combinations, we can easily expect them to act similarly. We assert that our segmentation method based on this inter-cluster measure can show a tendency similar to the min–max cut method, although we do not directly minimize Eq. (12) for partitioning. However, since this is not a focus of our paper, we omit experimental evaluations with detailed explanations.

4. Experimental results

In this section, we evaluate the proposed method by applying it to various kinds of data including image sequences as well as single frame data such as points and gray-scale, color, and texture images. In Section 4.1, we mainly describe the parameter selection of the affinity matrix and the computational cost of the proposed method. Section 4.2 presents main segmentation results and compares our results with those of the Ncut method and human segmentation for various images including

some from the Berkeley datasets. Some of these results were already presented (Jeon et al., 2005). In Section 4.3, we show that the proposed method can be applied to motion segmentation as well.

4.1. Parameter selection and computational cost

We have attempted to establish that our positiveness-based approach presents more superior and acceptable segmentation results than the Ncut method. To objectively evaluate the performance of both methods, we tried to perform the experiments under the same conditions.

First of all, the parameters such as the sigma for spatial adjacencies σ_X and the neighborhood size r are set to be the same in both methods; therefore, we can perform a fair comparison using the same affinity values. While the Ncut method uses a fixed value for the sigma for feature similarities σ_F, our approach automatically determines the parameter so that it is not only based on a cumulative value of the histogram but also robust to diverse input data. However, for most input images, the manually selected σ_F is not significantly different from the proposed automatically determined one.

Here, we explain how to compute the affinity matrix by automatically selecting parameters. Since these parameters determine the shape of the affinity matrix, they can significantly affect the performance of the proposed method. If we set them inappropriately, we cannot extract any segmentation information from the affinity matrix. Most of the previous spectral clustering methods manually selected these parameters depending on the input data, which makes it difficult to reproduce their algorithms. In this section, we present rules for the automatic determination of parameters, and experimentally show that they perform well for any kind of input data and that the segmentation results are not sensitive to small changes of these parameters. To determine σ_F in Eq. (1), we first normalize the norms of the feature vector differences, $\|\boldsymbol{F}(\boldsymbol{X}_i) - \boldsymbol{F}(\boldsymbol{X}_j)\|$, and make a histogram from them. Then, we set the σ_F as the value where the cumulative value of the histogram is about 20% of total histogram values – this procedure resembles the *noise estimator* described by Canny (1986). We also assert that the neighborhood size r and the sigma σ_X for the spatial adjacencies in Eq. (1) should be set to be large so that the feature similarities between data with fairly large distances are sufficiently expressed in the affinity matrix. If r is too large, the computation of the affinity matrix can be inefficient and spatial adjacency information cannot be well expressed. In contrast, if r is too small, the segmentation results can be fragmented and spatial information can be over-emphasized. Further, if σ_X can be appropriately determined in proportion to r, the shape of the affinity matrix will be good. In our experiments, if the number of pixels is N_p, then we set r to be approximately $0.8\sqrt{N_p}$ and σ_X to be about $0.6r$. By following these rules, we need not be bothered with setting many parameters and can obtain good and stable results.

The computational cost for the tensor factorization can be heavy. A 64×64 texture image, which corresponds to an affinity matrix of 4096×4096, can require 10 min to perform the initial PTF at the root node of the binary tree. Going down to the leaf nodes of the binary tree, this computational time is gradually reduced since the number of data is reduced. Further, we can significantly reduce the computational cost of the proposed PTF-based method by reducing the number of graph nodes which is actually the amount of data. This is possible by using over-segmentation: we can over-segment an original image into a lot of regions and regard them as new graph nodes instead of pixels. By using this approach, we can expect the proposed segmentation to be completed within several seconds.

4.2. Image segmentation

In this section, we compare the proposed method with human segmentation as well as the Ncut method which is a well-known clustering-based segmentation method. The results show that the proposed positiveness-based method is a good and acceptable tool for image segmentation.

4.2.1. Results of image segmentation and their analysis

Fig. 2 shows the segmentation result of a real texture image, from which we can understand how the proposed method performs unsupervised sparse clustering. As explained in Section 2.1, for the texture image such as Fig. 2 we first compute feature vectors using Gabor filters, where the window size is 7×7 and the numbers of scales and orientations are 5 $(3,4,5,6,7)$ and 6 $(0°, 30°, 60°, 90°, 120°, 150°)$, respectively. Next, we compute the affinity matrix from the Gabor responses and perform a hierarchical segmentation based on PTF. In Fig. 2(a), a binary tree structure is shown, where each node corresponds to a set of clustered pixels. In this figure, there are two kinds of numerals – under each node is its intra-cluster score, $h_1(C_i)$, and between two nodes is their inter-cluster score, $h_2(C_j, C_k)$. If $h_1(C_i)$ is larger than $T_1(=0.11)$, the cluster C_i is not factorized any more, and if $h_2(C_j, C_k)$ is larger than $T_2(=0.25)$, the factorized clusters C_j and C_k are removed. In Fig. 2(b), the original and a labeled-region image are shown in the first row, and three region images, each having one of the segmented regions, are shown in the second row. The labeled-region image is obtained by assigning the same gray-level to the pixels in the same segmented region, so it has the same number of gray-levels as segmented regions.

Since the evaluation of image segmentation results is subjective, it is difficult to define the only solution and thus the ability of an algorithm to provide multiple solutions is also important. In Fig. 3, we provide several segmentation results by changing thresholds for the intra- and inter-cluster measures, and compare them with those of the Ncut method. For this, we implemented the Ncut method using Gabor filters as well as *DOOG (Difference of Offset Gaussians) filters*, but only the results using the Gabor filters

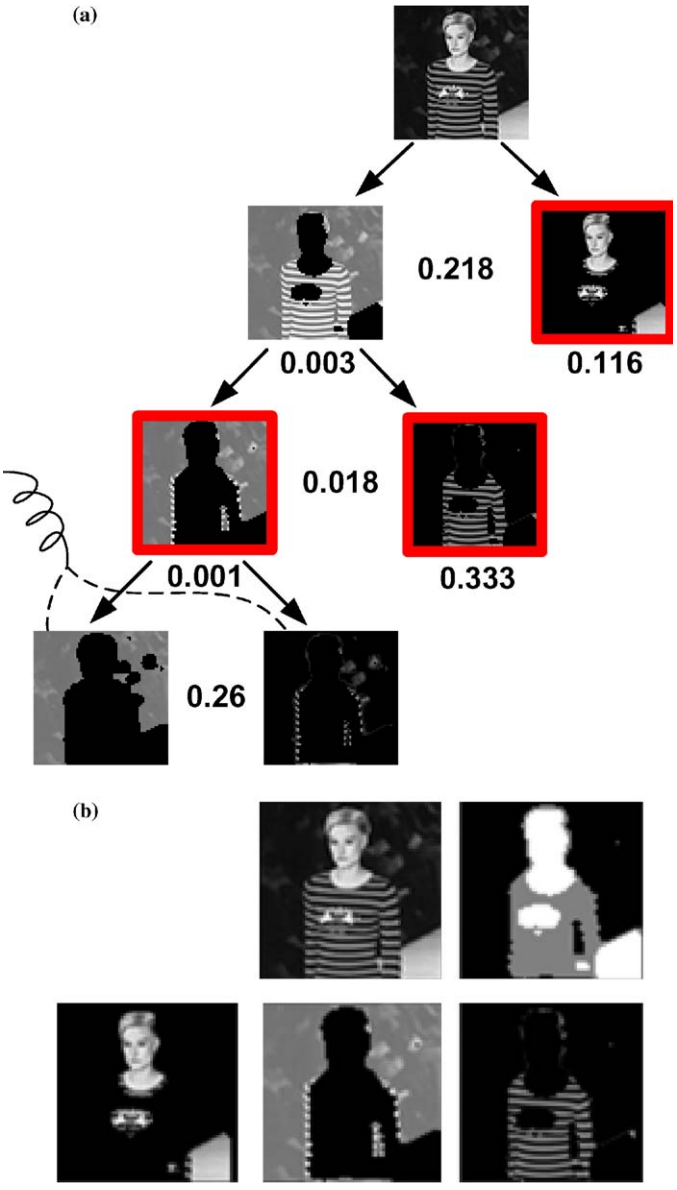

Fig. 2. Segmentation result of a real texture image ('fashion show', 69×69). (a) Segmentation procedure by the binary tree ($T_1 = 0.11, T_2 = 0.25$). (b) The original and a labeled-region image followed by three segmented regions.

are shown in Fig. 3(b). This is because the Ncut results using Gabor and DOOG filters are almost the same, and the comparison with our method should be performed under the same conditions. The Ncut method also performs a recursive bi-partitioning, which stops if the normalized cut value is larger than the threshold T_{Ncut}. The results of the proposed method are shown in Fig. 3(a), where the first column is from Fig. 2(b) and the second and third column are obtained by altering the values of T_1 and T_2 from those of the first one, respectively. In the second column, by increasing T_1 for the intra-cluster measure, we factorize more the first region of the first column into a homogeneous region with inner parts of the face and the stage and a textured boundary region enclosing them. In the third column, by increasing T_2 for the inter-cluster measure, we can obtain results more sensitive to soft textures in the background. As for the texture features, the textured hair and mark on the chest should be segmented from the human body. In Fig. 3(a), the proposed method successfully segments these regions from the human body as expected, although it fails to further segment them into two regions. This is because they look similar in the sense of Gabor filter responses, but this failure is not significant because we can easily split these parts using methods such as connected component labeling (CCL). The results of the Ncut method are shown in Fig. 3(b). In its first column, the Ncut method succeeds in separating the face and the stage into two regions, but fails in maintaining the homogeneous background in a single region. Furthermore, it does not separate the mark on the chest from the clothes although they are different textures. In the second and the third column, T_{Ncut} was increased to try to separate the mark, but this eventually caused unwanted over-segmentation. This can be prevented by the intra-cluster measure of the proposed method. These results show that the Ncut method can perform inaccurate segmentation because it only considers variance information related to eigenvectors.

In contrast to the Ncut method, the proposed method has an ability to prevent over- or under-segmentation by interactively using both intra- and inter-cluster measures. In the Ncut method, if we raise the threshold value of T_{Ncut} to factorize a cluster that should be factorized, the algorithm may factorize another cluster that should not be factorized, and vice versa. This is because there is not another measure to stop unwanted factorizations. In the proposed method, this unwanted segmentation can be easily prevented by first factorizing a cluster with a T_1 value, and next checking the factorized components once again with an appropriate T_2 value. We usually selected T_1 values sufficiently small so that all clusters that should be factorized could be factorized and selected T_2 values so that unwanted factorizations could be removed. In our experiments, we initially selected $(0.1, 0.25)$ as a set of the empirically tuned threshold values, T_1 and T_2, which could provide relatively good segmentation results. Then, we just varied this empirically tuned parameter set slightly so as to obtain the best segmentation results of the given images.

In case of the intra-cluster measure, eigengap, defined as $1 - \lambda_2/\lambda_1$, we can see that the farther λ_2 is from $\lambda_1(=1)$, the larger the eigengap gets, which means the given image loses the chance to be factorized as λ_2 decreases. In our experiments, we thought that when λ_2 is smaller than 0.9, the given image should not be factorized any more, so we usually selected 0.1 as the T_1 value. We observed that most images could be well segmented by the proposed method with T_1 values ranging from 0.05 to 0.2 according to the image contents. In the same way, to obtain the best segmentation results we can define the inter-cluster threshold value T_2 as a value ranging from 0.06 to 0.4, with a base value of 0.25, which can provide roughly good segmentation results. Further, as shown in Fig. 3(a), our method can provide various segmentation results according to user

Fig. 3. Comparison of the segmentation results. The result of each experiment using specific parameters is represented in each column where the segmented regions come after the labeled-region image shown in the first row. (a) Three results of the proposed method (first: $T_1 = 0.11$, $T_2 = 0.25$, second: $T_1 = 0.15$, $T_2 = 0.25$, third: $T_1 = 0.11$, $T_2 = 0.4$). (b) Two results of the Ncut method with our hierarchical structure (first: $T_{\text{Ncut}} = 0.235$, second: $T_{\text{Ncut}} = 0.25$).

requirements by changing thresholds T_1 and T_2. Therefore, we assert that the proposed method can provide superior segmentation results to the Ncut method by interactively using both the intra-cluster and the inter-cluster measures.

Fig. 4 shows the segmentation results of two synthetic point sets, and Fig. 5 shows a gray-scale and a color image, all of which seem to be simple and easy. While we do not directly compare our results with those of the Ncut approach, we can confirm that our PTF-based method is appropriate for segmentation in the same context as the previous explanation. Using simple synthetic data for the performance evaluation is a widely used method as we can easily realize the performance of the proposed method at a glance. Especially, from Fig. 5, we can see that our method performs well even for images that are very noisy or composed of thin and small regions.

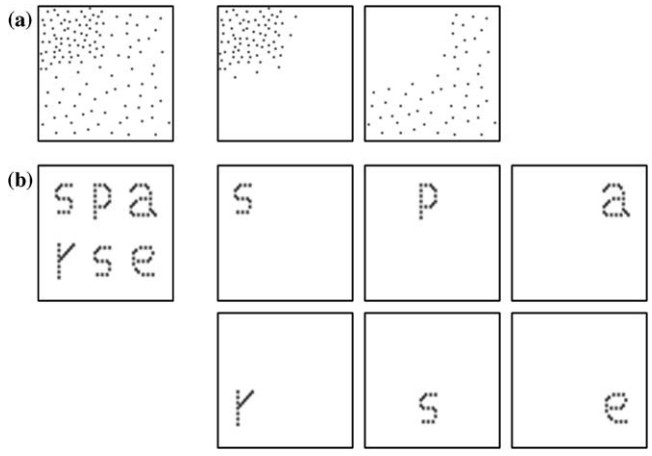

Fig. 4. Segmentation results of two point sets. (a) $T_1 = 0.11$, $T_2 = 0.25$. (b) $T_1 = 0.11$, $T_2 = 0.25$.

Fig. 6 shows the segmentation results of a real gray-scale image comparing our method with the Ncut method. The Ncut result is excerpted from the original paper of Shi and Malik (2000), where the parameters of $\sigma_F = 0.1$, $\sigma_X = 4.0$, and $r = 5$ were used for computing the affinity matrix. In Fig. 6(c), while the Ncut method provides relatively good segmentation results by using the eigenvectors under the orthogonality constraint, segmentation results in some regions do not coincide with our perception. Especially in the background, there is a clear boundary separating two regions with significantly different gray values, so human beings usually perceive this background as two regions. However, the Ncut method fails to partition the background into two regions because it relies only on the eigenvector representing the direction with the largest variance under the orthogonality constraint. In contrast to the Ncut method, in the first column of Fig. 6(b), the proposed method accurately segments this background

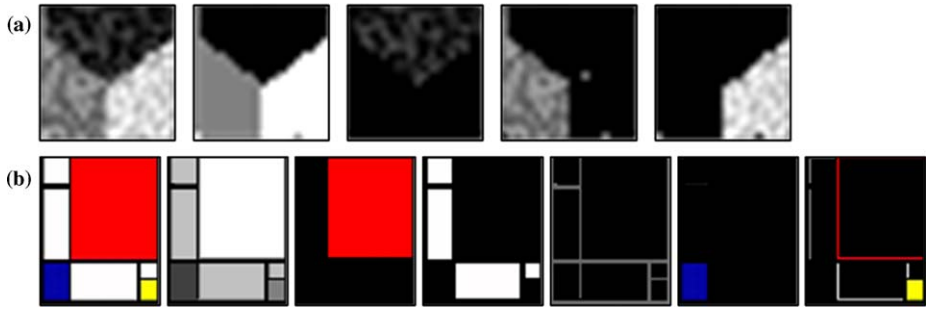

Fig. 5. Segmentation results of a synthetic gray-scale and a color image. In each result, the original and a labeled-region image are represented with segmented regions. (a) 'junction', 20×20 ($T_1 = 0.11$, $T_2 = 0.1$). (b) 'mondrian', 60×72 ($T_1 = 0.05$, $T_2 = 0.25$). (For interpretation of references in color, the reader is referred to the web version of this article.)

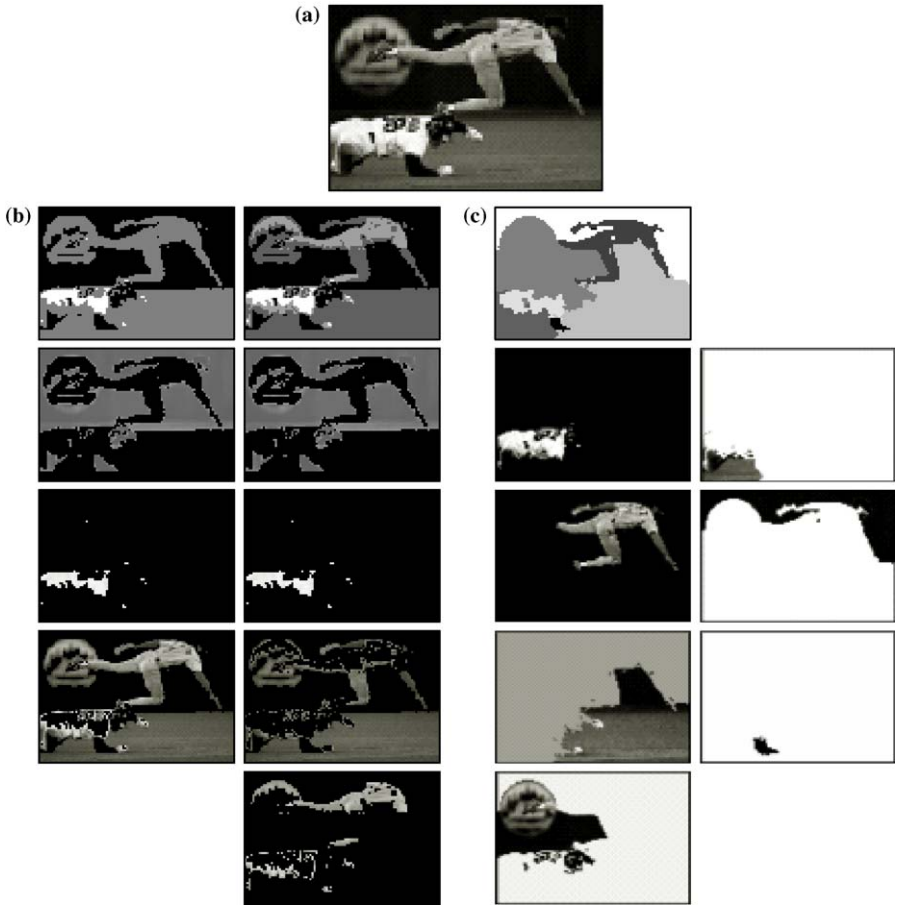

Fig. 6. Segmentation results of a real gray-scale image. In (b) and (c), the three results of the proposed and the Ncut method are shown in columns. In each result, a labeled-region image is followed by the segmented regions. (a) The original image ('baseball', 96×66). (b) Two results of the proposed method (first: $T_1 = 0.2$, $T_2 = 0.1$, second: $T_1 = 0.1$, $T_2 = 0.1$). (c) The result of the Ncut method (Shi and Malik, 2000) ($T_{\text{Ncut}} = 0.04$).

into two regions since this clear boundary can be easily discerned under the positiveness constraint. In the second column, we increase the threshold T_1 for the intra-cluster measure and obtain a more factorized player where the resulting boundaries are meaningful. Therefore, viewed holistically, our results, merging the upper human body with the background region, are not necessarily incorrectly segmented because human beings perceive the similarity between the upper human body and the background region in the sense of gray-levels. However, the Ncut method may present segmented regions which look bad or are different from the human segmented results. In other words, our method tends to find more reasonable segmentation results than the Ncut method with maximal separability.

Figs. 7 and 8 show that the proposed method also performs well in real color and synthetic texture images. The reason why we perform some experiments using simple synthetic data such as illustrated in Fig. 8 is that we can easily discern the performance of the proposed method qualitatively. In Fig. 8(a), we apply our method to the

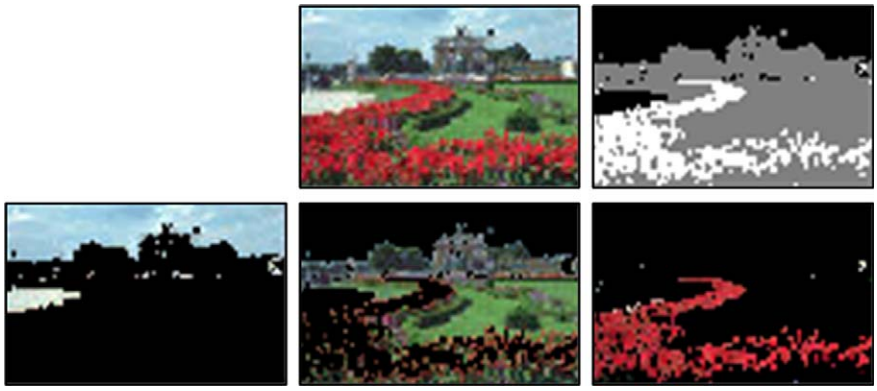

Fig. 7. Segmentation result of a real color image ('flower garden', 82×53). The original and a labeled-region image are represented with three segmented regions ($T_1 = 0.11, T_2 = 0.3$). (For interpretation of references in color, the reader is referred to the web version of this article.)

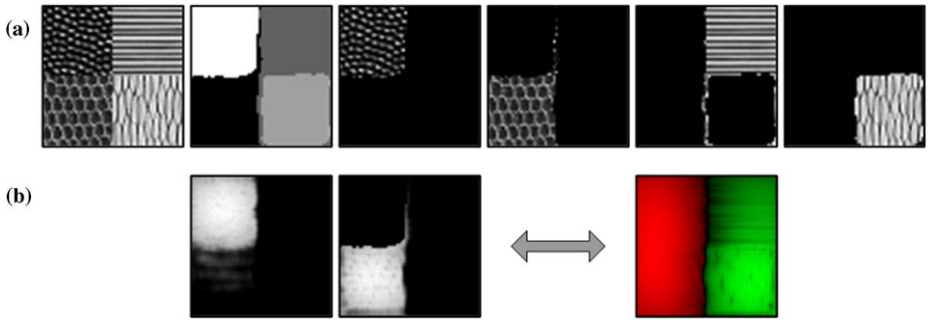

Fig. 8. Results of a texture image ('texture mosaic', 64×64). (a) The original and a labeled-region image followed by four segmented regions ($T_1 = 0.11, T_2 = 0.25$). (b) PTF results of the proposed method vs. the eigenvector of the Ncut method.

image of Brodatz-like texture patterns widely used for texture segmentation and obtain a satisfactory result. Fig. 8(b) represents the two factorized components by the PTF – two columns of $\boldsymbol{C}^{(1)}$ in Eq. (8) and the second smallest eigenvector by the Ncut method. They are computed at the root node of the binary tree where all the pixels of the original image are utilized. As explained in Section 2.3, the proposed and the Ncut method can perform segmentation by discretizing the components and the eigenvector, respectively. A careful observation of Fig. 8(b) shows the differences between the proposed and the Ncut method. In the components factorized by the PTF, most of entries are zero; therefore, the proposed method is called a *sparse* method. In contrast, almost all of the entries of the eigenvector have non-zero values, and some of them are even negative – they are indicated in red, which corresponds to the left half of the eigenvector image. The negative entries cause complex cancellations by subtractive combinations and lack *intuition* and *physical meaning*. Hence, we assert that our method based on *positiveness* is a significant advance in finding combinational components from positive data.

4.2.2. Experiments using the Berkeley datasets

To further justify our assertions, we performed additional experiments using the Berkeley dataset which provides human segmentation results of a great many images. The reason why many researchers use the datasets for testing or benchmarking their own algorithm is that the datasets provide not only vast input images for segmentation but also the results of the human segmentation obtained by many users. We can easily evaluate the performance of the proposed method by comparing it with human segmentation as well as the Ncut method.

In Fig. 9, we compare our method with the Ncut method for a real color image from the Berkeley datasets. From this, we can confirm that the results of the proposed method are close to those of human segmentation. Especially, as shown in the last row of Figs. 9(b) and (c), while the Ncut method often fails to provide clear boundaries, the proposed method successfully finds the boundaries of the building. It means that although the Ncut method is likely to provide more or less satisfactory segmentation results, it cannot present the elaborate regions such as object boundaries, which are easily recognized by humans. However, our PTF-based method exactly partitions the color image into the proper regions including boundaries. Further, the results of the proposed method tend to be similar to those of diverse human segmentation presented in Fig. 9(d). Therefore, we can claim that our method gives us more acceptable segmentation results.

Fig. 10 shows the segmentation results for a real texture image which can emphasize the advantages of the proposed

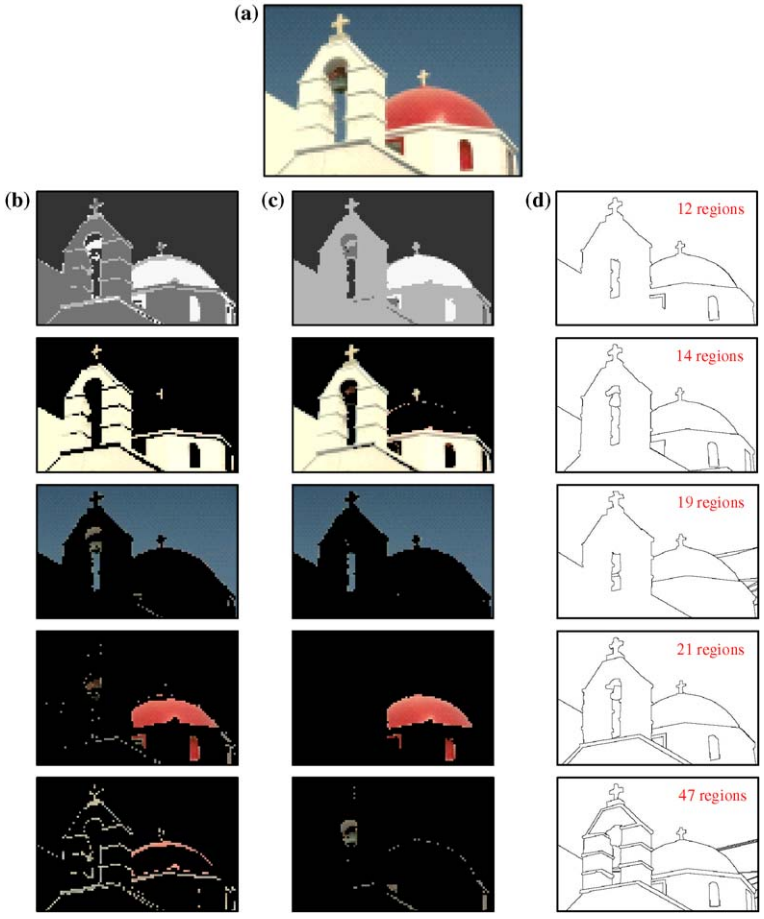

Fig. 9. Segmentation results of a real color image from the Berkeley datasets. In (b) and (c), the segmented regions of each method come after the labeled-region image shown in the first row. (a) The original image ('church', 96×64). (b) Result of the proposed method ($T_1 = 0.11, T_2 = 0.06$). (c) Result of the Ncut method with our hierarchical structure ($T_{\text{Ncut}} = 0.13$). (d) Results of the human segmentation from the Berkeley datasets. (For interpretation of references in color, the reader is referred to the web version of this article.)

method. In this figure, our method seems to coincide with multiple human segmentations, unlike the Ncut method. In Figs. 10(c) and (d), while our PTF-based method connected with the binary tree structure provides the proper segmentation result, the Ncut method excluding this structure does not correctly segment the image, especially the neck and shoulder. When our binary tree structure is incorporated, the Ncut result is improved, as shown in Fig. 10(e). However, this may be regarded as worse overall for not avoiding over-segmentation. In other words, while the Ncut method with our hierarchical structure is somewhat better, even the improved Ncut method does not surpass the proposed method since the eigenvector-based method does not have the advantages of our method: sparse clustering under the positiveness constraint and similar results to human segmentation.

Figs. 11 and 12 are other segmentation results which also show that the proposed method follows human segmentation well.

From the various experimental results, we assert that the positiveness and the hierarchical structure with two criteria can interactively improve the segmentation results, showing an inclination to be close to human segmentation. Furthermore, the hierarchical structure with two criteria can considerably, even though not perfectly, reduce the disadvantages of the Ncut method without utilizing the advantages of positiveness.

4.3. Motion segmentation

In this section, we show that the proposed method can be applied to other vision applications such as motion segmentation. For motion segmentation, we followed the same procedure as Fig. 1, which performed well in image segmentation, while using a different feature vector called *motion profile* (Shi and Malik, 2000). Let $I^t(\boldsymbol{X}_i)$ denote an image window centered at the pixel position $\boldsymbol{X}_i$ at time t and $P_i(\boldsymbol{dx})$ be the motion profile at $\boldsymbol{X}_i$ with displacement vector $\boldsymbol{dx}$ in the x and y directions. Then, $P_i(\boldsymbol{dx})$ is obtained by computing a *normalized SSD* between two image patches of $I^t(\boldsymbol{X}_i)$ and $I^{t+1}(\boldsymbol{X}_i + \boldsymbol{dx})$. If the feature distance between nodes i and j is denoted by d_{ij} and the spatial adjacency information is ignored in Eq. (1), the affinity between i and j is expressed as follows:

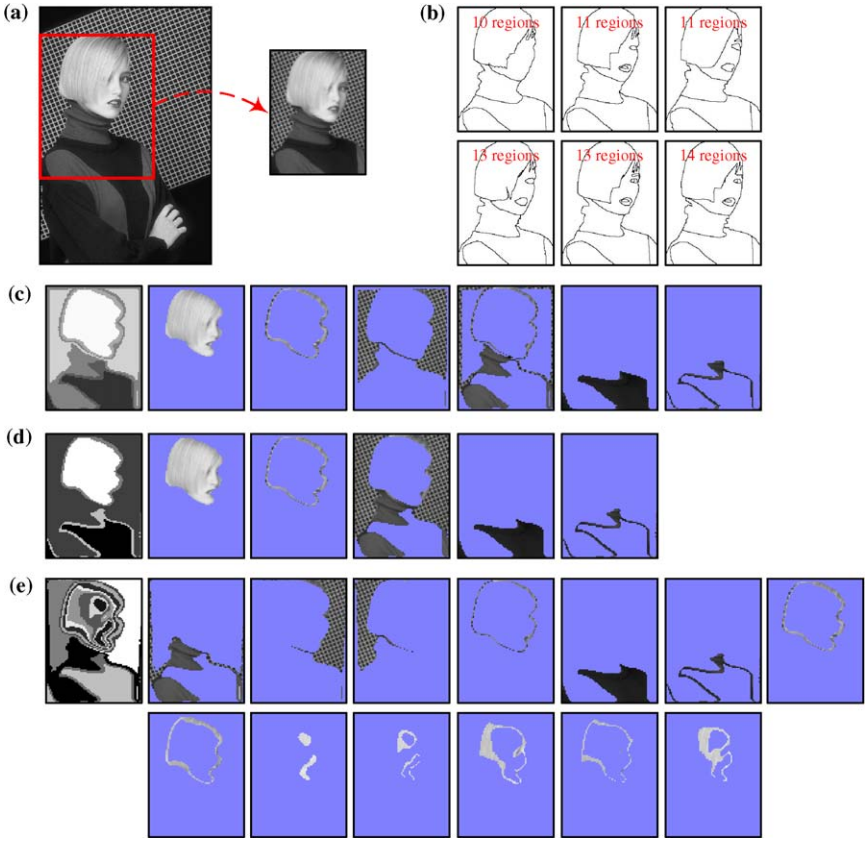

Fig. 10. Segmentation results of a real texture image from the Berkeley datasets. In (c) through (e), the three results of the proposed and the Ncut method are shown in rows. In each result, a labeled-region image is followed by the segmented regions. (a) The original image ('woman', 73×95). (b) Results of the human segmentation from the Berkeley datasets. (c) Result of the proposed method ($T_1 = 0.06, T_2 = 0.35$). (d) Result of the Ncut method without our hierarchical structure ($T_{\text{Ncut}} = 0.37$). (e) Result of the Ncut method with our hierarchical structure ($T_{\text{Ncut}} = 0.37$).

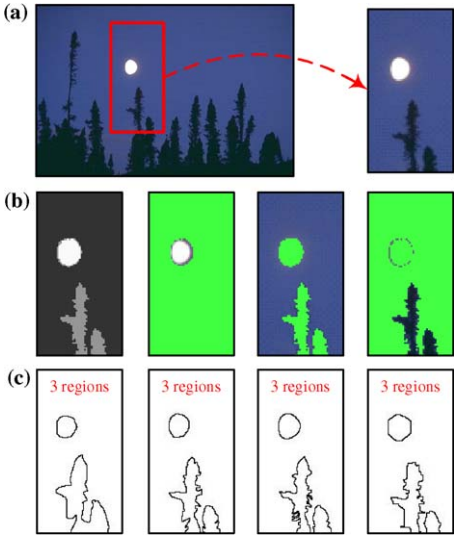

Fig. 11. Segmentation result of a real color image from the Berkeley datasets. In (b), a labeled-region image is followed by three segmented regions. (a) The original image ('moon', 50×96). (b) Result of the proposed method ($T_1 = 0.11, T_2 = 0.25$). (c) Results of the human segmentation from the Berkeley datasets. (For interpretation of references in color, the reader is referred to the web version of this article.)

$$A_{ij} = \mathrm{e}^{-d_{ij}^2/\sigma_{\mathrm{F}}^2}, \quad d_{ij} = 1 - \sum_{\boldsymbol{dx}} P_i(\boldsymbol{dx}) P_j(\boldsymbol{dx}), \tag{14}$$

where d_{ij} is one minus the cross-correlation of the motion profiles and nodes i and j are the pixels of the image at time t. If motion profiles between i and j are similar, they are probably grouped into the same motion segment.

Motion estimation is known to be an *ill-posed problem* due to uncertainties called the *aperture problem*. It means that we have difficulties in accurate motion estimation where there is only one-dimensional or no texture information. To alleviate this problem, we improve Eq. (14) by incorporating *contextual information* and the *smoothing-out property of the outlying pixels* into the affinity and the motion distance as follows:

$$\widehat{A}_{ij} = \begin{cases} W \times \overline{A}_{ij} & \text{if } \|\boldsymbol{V}_i - \boldsymbol{V}_j\| < T_v, \\ \overline{A}_{ij} & \text{otherwise,} \end{cases} \tag{15}$$

$$\overline{A}_{ij} = \mathrm{e}^{-\bar{d}_{ij}^2/\sigma_{\mathrm{F}}^2}, \quad \bar{d}_{ij} = \min\{d_{ij}, K \times \max\{d_{ij}\}\}, \tag{16}$$

where $\boldsymbol{V}_i$ is the intensity or the color at pixel i, W and T_v are the weight and the threshold for adding contextual information, and K is another weight for smoothing out the

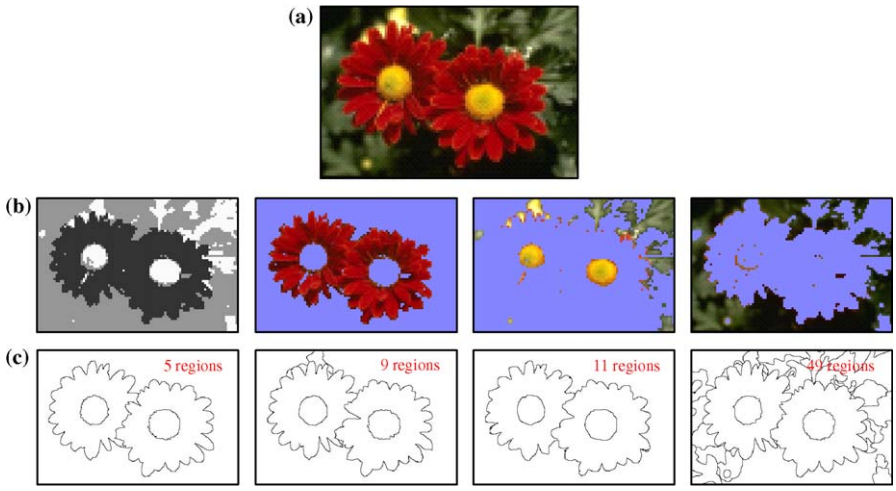

Fig. 12. Segmentation result of a real color image from the Berkeley datasets. In (b), a labeled-region image is represented with three segmented regions. (a) The original image ('flower', 96×64). (b) Result of the proposed method ($T_1 = 0.05, T_2 = 0.2$). (c) Results of the human segmentation from the Berkeley datasets. (For interpretation of references in color, the reader is referred to the web version of this article.)

outliers. Since the pixels with similar intensities or colors seem to have similar motion profiles, we increase the affinity between those pixels ($W = 2$), and this high-level information can improve our method. Further, we try to decrease the effect of outliers on the affinity by limiting the maximum value of motion distances using K in Eq. (16). Experimental results show that the use of this contextual information and outlier-smoothing property provides good performance.

Fig. 13 shows that the motion segmentation results of the proposed method and the Ncut method using a real gray-scale image sequence. The Ncut result of Fig. 13(c) is excerpted from the original paper which uses 5×5 image patches and ± 5 displacement vectors in x and y for computing motion profiles (Shi and Malik, 2000). Two results of the proposed method are shown in Fig. 13(b) using the same parameters as those of the Ncut method without consideration of spatial adjacency information. In this sequence, the moving object is the upper half of a man's body, but most parts of it have little texture information except the face. This lack of information can cause uncertainties and thus motion profiles can be inaccurate. If we apply the proposed method with the affinity function of Eq. (14), we just obtain the first column result in Fig. 13(b) where some parts of the moving body are wrongly segmented into the background. However, by using the improved distance and affinity function of Eqs. (15) and (16), we can distinguish almost all parts of the

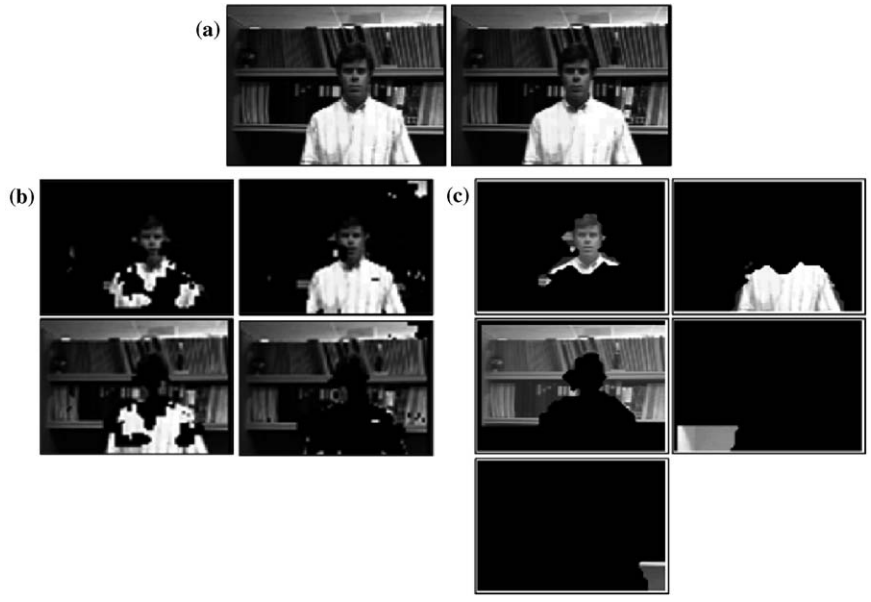

Fig. 13. Motion segmentation results using a real gray-scale image sequence. In (b) and (c), the three results of the proposed and the Ncut method are shown in columns. (a) The original image sequence. (b) Two results of the proposed method (first: $T_1 = 0.11$, $T_2 = 0.25$, second: $W = 2$, $T_v = 50$, $K = 0.9$, $T_1 = 0.11$, $T_2 = 0.25$). (c) The result of the Ncut method (Shi and Malik, 2000) ($T_{\text{Ncut}} = 0.06$).

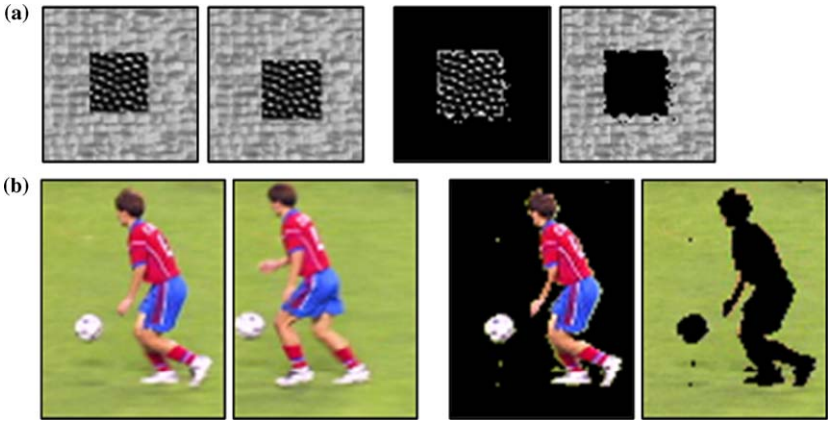

Fig. 14. Motion segmentation results using (a) a synthetic gray-scale and (b) real color image sequences. In each result, the first two images are the given image sequence and the other images are the motion segmented result which consists of a moving object and its background. (a) $T_1 = 0.11$, $T_2 = 0.14$. (b) $W = 2$, $T_v = 60$, $K = 0.8$, $T_1 = 0.11$, $T_2 = 0.25$. (For interpretation of references in color, the reader is referred to the web version of this article.)

moving body from the background. In Fig. 13(c), the Ncut method presents a less noisy segmentation result than our method, but it is over-segmented – it seems there are five regions moving differently from one another.

In Fig. 14, we present the motion segmentation results of synthetic gray-scale and real color image sequences. Since the synthetic sequence in Fig. 14(a) has sufficient texture information, we can obtain a good result despite using the standard affinity of Eq. (14). However, the real color sequence in Fig. 14(b) has two moving objects, and most of their parts are uniform regions. In this case, we can accurately segment the moving objects by using the improved distance and affinity functions.

5. Conclusion

In this paper, we presented a novel solution for the image segmentation problem by proposing a sparse clustering method based on PTF that tries to find additive combinations of positive components from the affinity matrix with originally positive entries. Further, by adopting a hierarchical structure of the binary tree with intra- and inter-cluster measures, we developed an unsupervised segmentation method which automatically determines the number of clusters. Our method provides interesting and challenging image segmentation results which show that *positiveness-based factorization* can have better performance than the method of optimizing graph cuts using eigenvectors under the orthogonality constraint. Moreover, to objectively evaluate the performance of our method, we compared the results of the proposed method with those of the human segmentation as well as the Ncut method for diverse input images including the Berkeley datasets. We also show that our method can be simply extended to motion segmentation, which is an important research area in computer vision and image processing. To our knowledge, the proposed method is the first segmentation approach based on positiveness and can be a new tool for vision applications. We believe our method will encourage many researchers who are interested in the human brain and perception.

The proposed method is the beginning phase of our research. We are planning several future works as follows. First, as mentioned in the preceding section, we will relieve the computational burden of our method through dimension reduction of the affinity matrix using over-segmentation methods. Second, we will develop and test a batch version of the proposed clustering method which is a k-way partitioning, not a *recursive bi-partitioning*. In spectral clustering, there is research showing that using more eigenvectors and directly computing k-way partitioning is better (Alpert and Yao, 1995). Next, we will consider a method of automatically determining the parameters which can give the best segmentation results according to the given image contents. In our method, we have provided empirically tuned threshold values and their small ranges of variations. Using them, the proposed method can give good segmentation results on a great many images. However, these empirical values cannot always guarantee good segmentation results for all kinds of images. Therefore, we are planning to find a more reliable and robust parameter determination method based on training approaches using information from a great many images. This approach would be intimately related to image classification, which is also a difficult research area. Finally, we will find good positive features for image segmentation and directly apply the proposed method to these features, not to the affinity matrix. The main issue of this approach is how to extract segmentation information from the positiveness-based factorizations of positive features. In this paper, we have not provided theoretical proof that positiveness is closely related to human perception since that would require greater knowledge about cognitive processes than is currently available. Therefore, we experimentally showed the advantages of our PTF-based method by comparing it with human segmentation as well as the Ncut method. If studies

on this topic become widespread, many researchers may have opportunities to find keys to the theoretical or the mathematical proof of our PTF-based method.

Acknowledgements

The authors would like to thank the *Ministry of Education of Korea* for financial support toward the Electrical and Computer Engineering Division at POSTECH through its *BK21 program*. They also thank the anonymous reviewers for their helpful comments, which improved the quality of this paper.

References

Adams, J., Bischof, L., 1994. Seeded region growing. IEEE Trans. Pattern Anal. Machine Intell. 16 (6), 641–647.

Alpert, C., Yao, S., 1995. Spectral partitioning: the more eigenvectors, the better. In: Proc. 32nd Design Automat. Conf., pp. 195–200.

Campbell, N.W., Thomas, B.T., Troscianko, T., 1997. Automatic segmentation and classification of outdoor images using neural networks. Internat. J. Neural Systems 8 (1), 137–144.

Canny, J., 1986. A computational approach to edge detection. IEEE Trans. Pattern Anal. Machine Intell. 8 (6), 679–698.

Chang, Y.L., Li, X., 1994. Adaptive image region growing. IEEE Trans. Image Process. 3 (6), 868–873.

Chennubhotla, C., Jepson, A., 2001. Sparse PCA – extracting multi-scale structure from data. In: Proc. IEEE Internat. Conf. Comput. Vision, pp. 641–647.

Cheriet, M., Said, J.N., Suen, C.Y., 1998. A recursive thresholding technique for image segmentation. IEEE Trans. Image Process. 7 (6), 918–920.

Chung, F., 1997. Spectral Graph Theory. American Mathematical Society.

Ding, C., He, X., Zha, H., Gu, M., Simon, H., 2001. A min–max cut algorithm for graph partitioning and data clustering. In: Proc. IEEE Internat. Conf. Data Mining, pp. 107–114.

Fiedler, M., 1975. A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory. Czech. Math. J. 25 (100), 619–633.

Geman, S., Geman, D., 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Machine Intell. 6 (6), 721–741.

Hagen, L., Kahng, A.B., 1992. New spectral methods for ratio cut partitioning and clustering. IEEE Trans. Comput.-Aided Design 11, 1074–1085.

Jain, A.K., Farrokhnia, F., 1991. Unsupervised texture segmentation using Gabor filters. Pattern Recognition 24 (12), 1167–1186.

Jeon, B., Jung, Y., Hong, K., 2005. Image segmentation by unsupervised sparse clustering. In: Proc. IEEE Workshop on Applications of Computer Vision (WACV/MOTION05), pp. 2–7.

Kannan, R., Vempala, S., Vetta, A., 2000. On clusterings – good, bad and spectral. In: Proc. 41st Symposium on the Foundation of Computer Science, pp. 367–377.

Lee, D.D., Seung, H.S., 1999. Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791.

Lovelock, D., Rund, H., 1989. Tensors, Differential Forms, and Variational Principles. Dover Publications, New York.

Lucchese, L., Mitra, S.K., 1999. Unsupervised segmentation of color images based on k-means clustering in the chromaticity plane. In: Proc. IEEE Workshop on Content-based Access of Images and Video Libraries, pp. 74–78.

Moore, A., 1999. Very fast EM-based mixture model clustering using multiresolution kd-trees. In: Proc. Adv. Neural Inform. Process. Systems, pp. 543–549.

Ng, A., Jordan, M., Weiss, Y., 2001. On spectral clustering: analysis and an algorithm. In: Proc. Adv. Neural Informat. Process. Systems, pp. 849–856.

Ohlander, R.B., 1975. Analysis of natural sciences. PhD Thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh.

Olshausen, B.A., Field, D.J., 1996. Emergence of simple-cell receptive-field properties by learning a sparse code for natural images. Nature 381, 607–609.

Paatero, P., Tapper, U., 1994. Positive matrix factorization – a nonnegative factor model with optimal utilization of error-estimates of data values. Environmetrics 5, 111–126.

Perkins, W.A., 1980. Area segmentation of images using edge points. IEEE Trans. Pattern Anal. Machine Intell. 2 (1), 8–15.

Prager, J.M., 1980. Extracting and labeling boundary segments in natural sciences. IEEE Trans. Pattern Anal. Machine Intell. 2 (1), 16–27.

Shi, J., Malik, J., 2000. Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Machine Intell. 22 (8), 888–905.

Strang, G., 1988. Linear Algebra and Its Applications. Harcourt Brace Javanovich, Inc., Orlando.

Welling, M., Weber, M., 2001. Positive tensor factorization. Pattern Recognition Lett. 22 (12), 1255–1261.

Wertheimer, M., 1938. Laws of organization in perceptual forms. In: Ellis, W. (Ed.), A Sourcebook of Gestalt Psychology. Routledge & Kegan Paul, London, pp. 71–88 (first published: 1923. Untersuchungen zur Lehre von der Gestalt II. Psycologische Forschung 4, 301–350).