Image Segmentation using a Weighted Kernel PCA Approach to Spectral Clustering

Carlos Alzate, Johan A. K. Suykens

***Abstract*— In classical graph-based image segmentation, a data-driven matrix is constructed representing similarities between every pair of pixels. The eigenvectors of such matrices contain relevant information about the clusters present on the image. An approach to image segmentation using spectral clustering with out-of-sample extensions is presented. This approach is based on the weighted kernel PCA framework. An advantage of the proposed method is the possibility to train and validate the clustering model on subsampled parts of the image to be segmented. The cluster indicators for the remaining pixels can then be inferred using the out-of-sample extension. This subsampling scheme can be used to reduce the computation time of the segmentation. Simulation results with grayscale and color images show improvements in terms of computation times together with visually appealing clusters.**

I. Introduction

Image segmentation consists of partitioning an image into several related regions. It is a fundamental problem in image analysis, computer vision, object recognition and is closely related to perceptual grouping. Several approaches from different perspectives have been developed to tackle the image segmentation problem: point-based methods, region-based algorithms, connectivity-preserving techniques and optimization-based methods. Point-based methods correspond to the simplest algorithms and are based on thresholding. Therefore, these methods are useful only when the contrast between the objects and the background is high enough. Region-based algorithm use merge and split techniques to create clusters based on some criterion such as homogeneity and compactness. Connectivity-preserving methods such as active contours and deformable models start with some preliminary shape and iteratively shrinks/expands it according to some energy function. Optimization-based methods such as neural networks, Bayesian learning and clustering algorithms aim to create partitions in such a way that some cost function is minimized. Typical optimization-based methods are k-means, EM clustering and graph-based clustering). Graph-based image segmentation methods have been proven to deliver good segmentation results using low-level image features such as color, texture and position [17], [14], [15], [20].

This paper focus on spectral clustering techniques which are the typical representatives of graph-based clustering. These methods are relaxations of graph partitioning problems that are generally NP-hard. The relaxed solutions that spectral clustering provides correspond to the eigenvectors of some data-driven similarity matrix containing pairwise similarities. When only two clusters are required, the bi-partitioning can be obtained by thresholding the eigenvector solution. However, the clustering problem becomes complicated when more than two clusters are required because it is not clear how to convert the relaxed solutions back to cluster indicators. For this purpose, several approaches from different points of view have been proposed [17], [3], [7]. One issue of image segmentation based on spectral clustering is the size of the similarity matrix. Every pixel in the image becomes a data point and pairwise similarities with all the other pixels have to be computed. These similarity matrices can be very large even for small images and the resulting eigenvalue decomposition becomes prohibitely slow. Another issue with spectral clustering is that the obtained clusters are defined only for the training data with no straightforward extension to out-of-sample (test) points.

Classical kernel PCA was introduced in [16] as a nonlinear generalization of PCA by using kernels. It was shown in [2] that several spectral clustering techniques can be formulated in terms of weighted kernel PCA. This formulation provides a unifying view of spectral clustering with primal/dual insights in a clear optimization framework. Therefore, extensions to out-of-sample data points can be calculated in an exact way by using the projections over the eigenvectors. In this paper, we take advantage of the out-of-sample extension provided by the weighted kernel PCA framework to perform image segmentation in a fast and efficient way. The proposed clustering method is first trained and validated on subsets of the image to be segmented and then the clusters for the remaining pixel subsets are inferred using the out-of-sample extension.

This paper is organized as follows. Section II contains a review of spectral clustering techniques. In Section III, we describe the weighted kernel PCA approach and the links to spectral clustering. In Section IV, we formulate the out-of-sample extension together with the subsampling scheme. Section V contains the empirical results and in Section VI we give the concluding comments.

II. Spectral Clustering

In spectral clustering, a set of data points can be represented as an undirected graph $G = (\mathcal{V}, \mathcal{E})$. The vertices of the graph $\mathcal{V}$ are the points and the edges $\mathcal{E}$ have an associated weight representing the degree of similarity between every pair of vertices. The problem of clustering consists of finding a partition of the graph such that the points within a cluster have high similarity and points in different clusters have low

The authors are with the Department of Electrical Engineering ESAT-SCD-SISTA, Katholieke Universiteit Leuven, B-3001 Leuven, Belgium (email: carlos.alzate@esat.kuleuven.be; johan.suykens@esat.kuleuven.be).

similarity. The simplest way to bipartition the graph into sets $\mathcal{A}, \mathcal{B}$ such that $\mathcal{A} \cap \mathcal{B} = \emptyset, \mathcal{A} \cup \mathcal{B} = \mathcal{V}$ is to minimize the *cut* which is the sum of the similarities that have to be removed to create two disjoint sets. The *cut* is defined as:

$$\text{cut}(\mathcal{A}, \mathcal{B}) = \sum_{a \in \mathcal{A}, b \in \mathcal{B}} s(a, b)$$

where $s(a, b)$ is the similarity between vertices a and b.

The problem of minimizing the *cut* is formulated as follows [9]:

$$\begin{aligned} \min_q J_{mincut} &= q^T(D - S)q \\ \text{such that} &\quad q \in \{-1, 1\}^N \end{aligned} \tag{1}$$

where q is a cluster indicator vector with i-th entry:

$$q_i = \begin{cases} 1, & \text{if } i \in \mathcal{A} \\ -1, & \text{if } i \in \mathcal{B}, \end{cases}$$

$D = \text{diag}(d_1; \ldots; d_N)$ is the degree matrix, $d_i = \sum_j s(i, j)$ and S is the similarity matrix with ij-th entry $S_{ij} = s(i, j)$.

Minimizing the *cut* is NP-hard due to the combinatorial constraint on q. A suboptimal solution can be found by relaxing the constraint and letting q take real values. The solution to the relaxed problem with constraint $\tilde{q}^T\tilde{q} = 1$ is given by $\tilde{q}^{(N-1)}$ the eigenvector corresponding to the second smallest eigenvalue of:

$$L\tilde{q} = \lambda\tilde{q}$$

where L is the unnormalized Laplacian defined as $L = D - S$ and $\tilde{q} \in \mathbb{R}^N$ is the relaxed cluster indicator. The ordered eigenvalues of L are $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N = 0$ with corresponding eigenvectors $\tilde{q}^{(1)}, \tilde{q}^{(2)}, \ldots, \tilde{q}^{(N)} = 1_N$. The relaxed solution needs to be converted back to a cluster indicator vector by thresholding:

$$q_i = \text{sign}(\tilde{q}_i - \theta), i = 1, \ldots, N.$$

The threshold θ can be found using a grid search and choosing θ such that (1) is minimized. In general, minimizing the *cut* leads to very imbalanced partitions because the size of the clusters is not taken into account.

The normalized cut (*NCut*) introduced in [17] penalizes small sets by taking into account the total weight of each cluster. Its relaxed solution follows from the following generalized eigenvalue problem:

$$L\tilde{q} = \lambda D\tilde{q}. \tag{2}$$

As in the *cut*, the relaxed solution is the eigenvector corresponding to the second smallest eigenvalue of (2).

A. K-Way NCut Relaxation

A more general problem consists of partitioning the graph into k disjoint sets $\mathcal{A}_1, \ldots, \mathcal{A}_k, k > 2$. The k-way *Ncut* is defined as:

$$\text{NCut}(\mathcal{A}_1, \ldots, \mathcal{A}_k) = \sum_{i=1}^{k} \frac{\text{cut}(\mathcal{A}_i, \overline{\mathcal{A}}_i)}{\text{Vol}(\mathcal{A}_i)}$$

where $\text{Vol}(\mathcal{A}) = \sum_{i \in \mathcal{A}} d_i$ is the volume of $\mathcal{A}$ and $\overline{\mathcal{A}}$ denotes the complement of $\mathcal{A}$.

Consider $f^{(k)} \in \{0, 1\}^N$ as the cluster indicator vector for the k-th cluster such that $f^{(k)}$ has a 1 in the entries corresponding to the data points in the k-th cluster. The cluster indicator matrix becomes $F = [f^{(1)}, \ldots, f^{(k)}]$. The k-way *NCut* can be formulated as

$$\text{NCut}(\mathcal{A}_1, \ldots, \mathcal{A}_k) = k - \text{tr}(G^T\hat{L}G)$$

where $\hat{L} = D^{-1/2}SD^{-1/2}$ is the normalized Laplacian, $G = [g^{(1)}, \ldots, g^{(k)}]$, $g^{(i)} = D^{1/2}f^{(i)}/||D^{1/2}f^{(i)}||_2, i = 1, \ldots, k$.

Relaxing the discrete constraint by allowing G to take real values leads to the k-way *NCut* relaxation [11]:

$$\begin{aligned} \max_{\tilde{G}} \text{NCut}(\mathcal{A}_1, \ldots, \mathcal{A}_k) &= \text{Tr}(\tilde{G}^T\hat{L}\tilde{G}) \\ \text{such that} &\quad \tilde{G}^T\tilde{G} = I_k \end{aligned}$$

This maximization problem can be solved using the Ky Fan's theorem [8],[5],[6]. The solution for this relaxed problem is given by

$$\tilde{H}^* = UR_1 \tag{3}$$

where $U \in \mathbb{R}^{N \times k}$ is any orthonormal basis of the k-th principal subspace of $\tilde{L}$ and $R_1 \in \mathbb{R}^{k \times k}$ is an arbitrary orthogonal matrix [8].

B. From Eigenvectors to Clusters

The solution provided by (3) is a real-valued matrix which does not correspond to cluster indicators. The problem of converting the eigenvectors back to cluster indicators is not straightforward and several approaches have been proposed. [17], [7], [10]. The classical approach is to perform k-means on the eigenvectors. This approach is called *reclustering* and was proposed in [17]. In [7], the problem of converting eigenvectors was translated to finding peaks and valley of a 1-D quantity called cluster crossing. A different approach was introduced in [3] which proposed looking for a cluster indicator matrix T that minimizes the difference between the subspaces spanned by T and U.

Using k-means as a technique for reclustering might not be appropiate because the eigenvectors are generally aligned forming lines and k-means assumes spherical clusters. This motivates the use of different reclustering algorithms such as k-lines [10]. A summarized description of the k-lines algorithm introduced in [10] is presented. Given a set of input variables $\{x_i\}_{i=1}^N, x_i \in \mathbb{R}^k$ and the number of clusters k, the algorithm returns a set of k points $\mathcal{M} = \{m_1, \ldots, m_k\}$ representing lines passing through the origin.

Algorithm 1 K-lines. Each cluster is represented by a vector $m_i \in \mathbb{R}^k$ of unitary norm $i = 1, \ldots, k$

1: Initialize $m_i, i = 1, \ldots, k$ (e.g. randomly)
2: For each $i \in \{1, \ldots, k\}$, let S_i be the set of points containing all points x_j that are closest to the line defined by m_i
3: For each $i \in \{1, \ldots, k\}$, let m_i define the line through the origin which is closest to all points in S_i
4: Repeat from 2 until convergence

III. Weighted Kernel PCA Framework

A. LS-SVM Formulation to Kernel PCA

An LS-SVM approach to kernel PCA was introduced in [18], [19]. This approach showed that kernel PCA is the dual solution to a primal optimization problem formulated in a kernel induced feature space. The underlying loss function associated to kernel PCA was shown to be L_2.

Given a set of N data points $\{x_i\}_{i=1}^N, x_i \in \mathbb{R}^d$, kernel PCA aims to find directions in which the projected variables $e = \Phi_c w$ have maximal variance along with minimizing a regularization term. This optimization problem can be formulated in the primal as:

$$\begin{array}{rcl} \max_{w,e} J_p(w,e) & = & \gamma \frac{1}{2} e^T e - \frac{1}{2} w^T w \\ \text{such that} & & e = \Phi_c w \end{array} \tag{4}$$

where Φ_c is the $N \times n_h$ feature matrix:

$$\Phi_c = \begin{bmatrix} \varphi(x_1)^T - \hat{\mu}_\varphi^T \\ \varphi(x_2)^T - \hat{\mu}_\varphi^T \\ \vdots \\ \varphi(x_N)^T - \hat{\mu}_\varphi^T \end{bmatrix},$$

γ is the regularization parameter, $\hat{\mu}_\varphi = (1/N)\sum_{i=1}^N \varphi(x_i)$ and $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d_h}$ is the mapping to a high dimensional feature space of dimension d_h.

The Lagrangian for this problem becomes:

$$\mathcal{L}(w, e; \alpha) = \frac{\gamma}{2} e^T e - \frac{1}{2} w^T w - \alpha^T (e - \Phi_c w)$$

with conditions for optimality given by:

$$\begin{array}{rcl} \frac{\partial \mathcal{L}}{\partial w} = 0 & \to & w = \Phi_c^T \alpha \\ \frac{\partial \mathcal{L}}{\partial e} = 0 & \to & \alpha = \gamma e \\ \frac{\partial \mathcal{L}}{\partial \alpha} = 0 & \to & e = \Phi_c w. \end{array}$$

Defining $\lambda = 1/\gamma$ and expressing the conditions for optimality only in terms of α leads to:

$$\Phi_c \Phi_c^T \alpha = \lambda \alpha.$$

By making use of the kernel trick $\varphi(x)^T \varphi(y) = K(x, y)$, the following dual eigenvalue problem is obtained:

$$\Omega_c \alpha = \lambda \alpha$$

where Ω_c is the centered kernel matrix $\Omega_c = U\Omega U$, U is the centering matrix $U = I_N - (1/N)1_N 1_N^T$, I_N is the $N \times N$ identity matrix, 1_N is a vector of N ones and Ω is the kernel matrix with ij-entry: $\Omega_{ij} = K(x_i, x_j)$.

The projection of a test data point x (also called score variable) onto the i-th eigenvector $\alpha^{(i)}$ becomes:

$$z_i(x) = w^T \varphi(x) = \sum_{l=1}^N \alpha_l^{(i)} K(x_l, x). \tag{5}$$

B. Weighted Kernel PCA

A generalized formulation for kernel PCA was proposed in [1]. This formulation was used to impose robustness and sparseness in kernel PCA by introducing a weighting matrix and choosing the weights in such a way that outliers have less influence than the rest of the data. The weighted kernel PCA formulation has also been used as a unifying framework for spectral clustering methods [2]. Classical spectral clustering algorithms such as the *NCut*, the random walks method, kernel alignment and the NJW algorithm were shown to be particular cases of weighted kernel PCA.

Extending (4) to a symmetric positive definite weighting matrix $V \in \mathbb{R}^{N \times N}$

$$\begin{array}{rcl} \max_{w,e} J_p(w,e) & = & \gamma \frac{1}{2} e^T V e - \frac{1}{2} w^T w \\ \text{such that} & & e = \Phi w \end{array} \tag{6}$$

with $\Phi = [\varphi(x_1)^T; \ldots; \varphi(x_N)^T], V = V^T > 0$.

The Lagrangian for this constrained optimization problem becomes:

$$\mathcal{L}(w, e; \alpha) = \frac{\gamma}{2} e^T V e - \frac{1}{2} w^T w - \alpha^T (e - \Phi w)$$

with conditions for optimality given by:

$$\begin{array}{rcl} \frac{\partial \mathcal{L}}{\partial w} = 0 & \to & w = \Phi^T \alpha \\ \frac{\partial \mathcal{L}}{\partial e} = 0 & \to & \alpha = \gamma V e \\ \frac{\partial \mathcal{L}}{\partial \alpha} = 0 & \to & e = \Phi w. \end{array}$$

By elimination of w and e, the following dual eigenvalue problem is obtained:

$$V \Omega \alpha = \lambda \alpha \tag{7}$$

and (5) still holds.

If the weighting matrix V is chosen to be the inverse of the degree matrix D then weighted kernel PCA corresponds to the *NCut*. Links with other spectral clustering methods for graph bipartitioning were shown in [2].

IV. Out-of-Sample Extension

The main advantage of the weighted kernel PCA formulation to spectral clustering is the possibility to apply the trained clustering model to out-of-sample (test) points. This is not possible in the classical formulation to spectral clustering, because there is no underlying model and the clusters are defined only for training points. Extensions to out-of-sample points for several unsupervised learning techniques including spectral clustering were discussed in [4]. These extensions rely on the Nyström method which is an approximation of the underlying eigenfunction. On the other hand, the weighted kernel PCA framework provides an exact way to extend the model to out-of-sample points without relying on approximations.

Given a set of N_{test} test points $\{x_j^{\text{test}}\}_{j=1}^{N_{\text{test}}}$, the score variables corresponding to the i-th eigenvector become:

$$z_i = \Phi_{\text{test}} \Phi^T \alpha^{(i)}$$

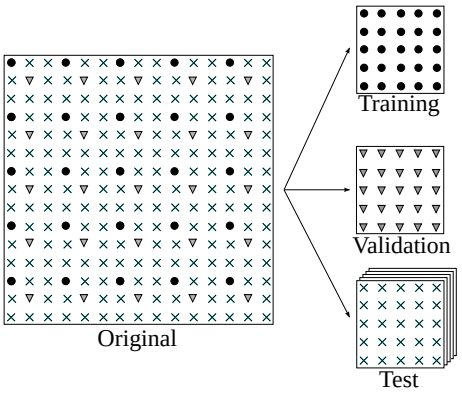


Fig. 1. Subsampling scheme

where Φ_{test} is the $N_{\text{test}} \times n_h$ feature matrix:

$$\Phi_{\text{test}} = \begin{bmatrix} \varphi(x_1^{\text{test}})^T \\ \varphi(x_2^{\text{test}})^T \\ \vdots \\ \varphi(x_{N_{\text{test}}}^{\text{test}})^T \end{bmatrix}.$$

Applying the kernel trick leads to

$$z_i = \Omega_{\text{test}}\alpha,$$

where $\Omega_{\text{test}} = \Phi_{\text{test}}\Phi^T$ is the $N_{\text{test}} \times N$ kernel matrix evaluated using the test points with jk-entry $\Omega_{\text{test},jk} = K(x_j^{\text{test}}, x_k)$.

The score variables matrix becomes $Z = [z_1, \ldots, z_l]$ where l is the size of the subspace spanned by the eigenvectors.

A. From Score Variables to Clusters

To obtain cluster indicators from the score variables, the k-lines reclustering approach over the score variables Z was used. Note that in the original k-lines algorithm, the input data are eigenvectors that lie in k-dimensional space, where k is the number of clusters to be found. We use score variables lying in $k-1$ dimensional space because the first principal direction already gives a binary partition. Therefore to obtain k clusters, only the score variables corresponding to the $k-1$ dominant eigenvectors have to be calculated.

B. Subsampling Scheme

The image to be segmented is first subsampled and several smaller images are obtained from the original. The clustering model is then trained on one of those smaller images. Model selection can be done by optimizing the parameters on a different subsampled part of the original image. The obtained model is then applied to the test set which corresponds to the pixels that were not used in the training and validation procedures. Figure 1 shows an illustration of the subsampling procedure. The last step consists of putting all the cluster indicators of the subsampled parts back into their positions in the original image.

V. Empirical Results

In this section, experimental results are shown. We used one grayscale image and one color image from the Berkeley image dataset [13]. For all the experiments, we subsampled the original image leaving 3 pixels out. This leads to a reduction of the number of pixels of the original image by a factor of 16. Shifting the subsampled image to cover all the original image leads to 16 possible combinations of shifted images. The first combination is used for training, the second for validation and the remaining 14 are test images for which the cluster indicators are inferred using the out-of-sample extension. . The edge similarity was calculated using a composite RBF kernel function [17]. If

$$||X(x_i) - X(x_j)||_2 < r$$

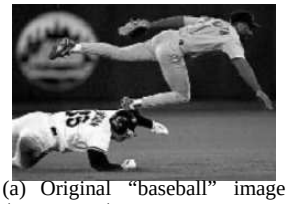
then the edge similarity becomes:

$$S_{ij} = e^{-\left(\frac{||F(x_i)-F(x_j)||_2^2}{\sigma_F^2} + \frac{||X(x_i)-X(x_j)||_2^2}{\sigma_X^2}\right)} \quad (8)$$

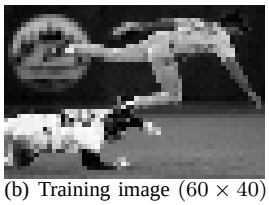
otherwise $S_{ij} = 0$, where $F(x_i)$ represents the feature vector for the point x_i (e.g. intensity for grayscale images, RGB or HSV representations for color images) and $X(x_i)$ denotes the spatial position of x_i. The σ_F^2 parameter was tuned using k-lines , and $r = 0.8\sqrt{N_p}$, $\sigma_X^2 = 0.6r$ where N_p is number of pixels of the image as proposed in [12].

A. Grayscale Segmentation

We report results on the classical "baseball" image introduced in [17]. Segmentation results are shown in Figure 2. The obtained clusters are visually and perceptually more appealing than the original *Ncut* algorithm. Figure 3 shows the model selection plots indicating that the k-lines criterion can be used for parameter tuning and as a reclustering approach.



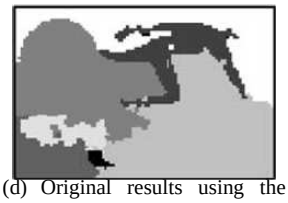
(a) Original "baseball" image (240×160)



(b) Training image (60×40)



(c) Labeled-region image using the proposed method



(d) Original results using the *NCut* as shown in [17]

Fig. 2. Grayscale segmentation - Training, validation and test scenario obtained by subsampling.

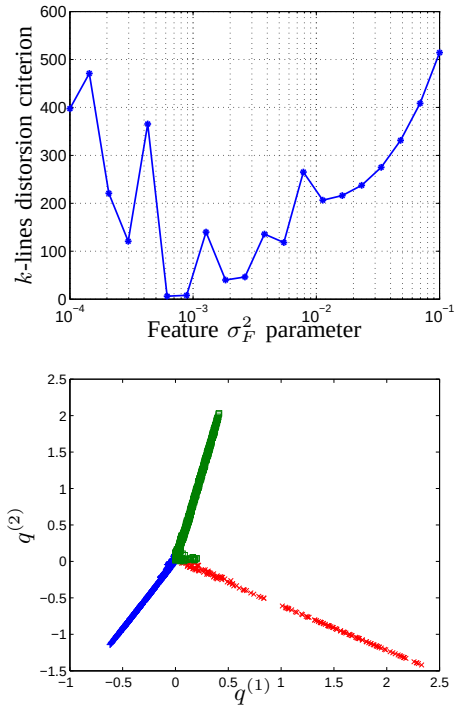


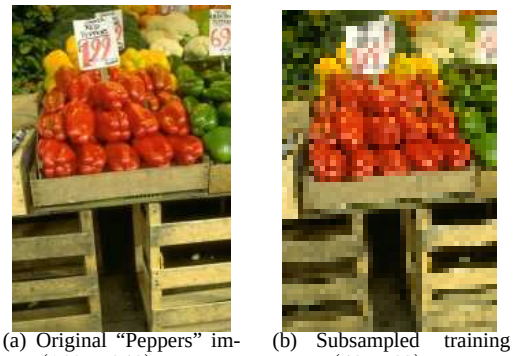
Fig. 3. Grayscale segmentation - Model selection. The top plot shows the feature σ^2 parameter tuning using the k-lines distorsion criterion. The obtained parameter is $\sigma_F^2 = 7 \times 10^{-4}$. The bottom plot shows the scores variables for the validation set. Note that the three clusters are strongly aligned forming lines.

B. Color Segmentation

For color segmentation, we used the "peppers" image from the Berkeley image dataset [13]. Figure 4 shows the original and subsampled training image. The model selection plots showed similar behavior to the grayscale segmentation experiment. The tuned σ_F^2 was 1×10^{-3}. Figures 5 and 7 show the segmentation results compared with a human segmentation taken from the Berkeley image dataset. The first segmented region comprises the background of the image while the second region groups the wooden boxes that correspond to the foreground. The third and fourth segmented regions correspond to detailed information of the image. The cluster boundaries are comparable to the boundaries found by the human segmentation. Figure 6 shows the computation times of the standard *NCut* and the out-of-sample extension relative to the number of pixels in the image. The experiment was carried out using Matlab on a Pentium 4, 2.8 GHz, 1 GB RAM.

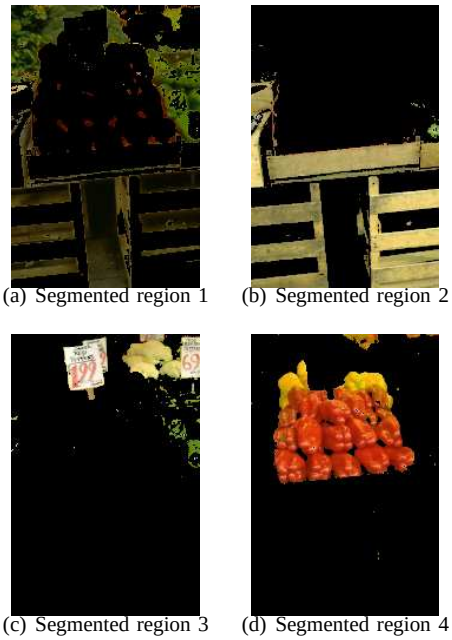
VI. Conclusions

An image segmentation method based on the weighted kernel PCA approach to spectral clustering is derived. An advantage of this approach is the possibility to infer cluster indicators for test pixels using the out-of-sample extension. This extension is characteristic of the weighted kernel PCA framework and is based on primal/dual insights. This leads to a reduction in the algorithm computation times by a factor of at least 8 compared to the standard *NCut* using the full set of pixels. The obtained clusters were obtained in fast way and the segmented regions were comparable to human segmentation. The k-lines algorithm was empirically shown to deliver good results for model selection and reclustering. This is due to the fact that the eigenvectors and score variables are aligned forming lines when relevant and visually appealing clusters are present on the image.



(a) Original "Peppers" image (160×240) (b) Subsampled training image (40×60)

Fig. 4. Color segmentation - "Peppers" dataset. Original and training images.



(a) Segmented region 1 (b) Segmented region 2

(c) Segmented region 3 (d) Segmented region 4

Fig. 5. Color segmentation - "Peppers" dataset - Segmentation results. The segmented regions show the background, foreground and detailed information of the image.

Acknowledgment

This work was supported by grants and projects for the Research Council K.U.Leuven (GOA-Mefisto 666, GOA-Ambiorics, several PhD / Postdocs & fellow grants), the Flemish Government FWO: PhD / Postdocs grants, projects G.0240.99, G.0211.05, G.0407.02, G.0197.02, G.0080.01, G.0141.03, G.0491.03, G.0120.03, G.0452.04, G.0499.04, G.0226.06, G.0302.07, ICCoS,

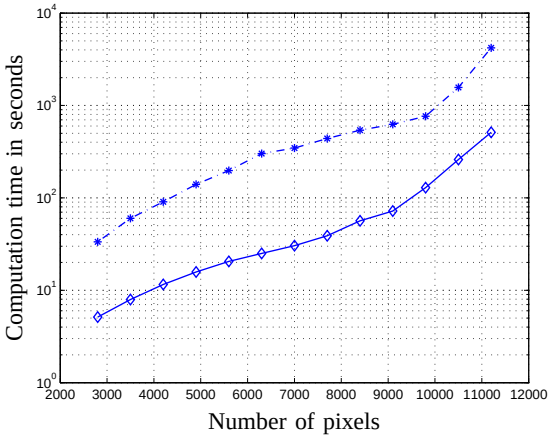
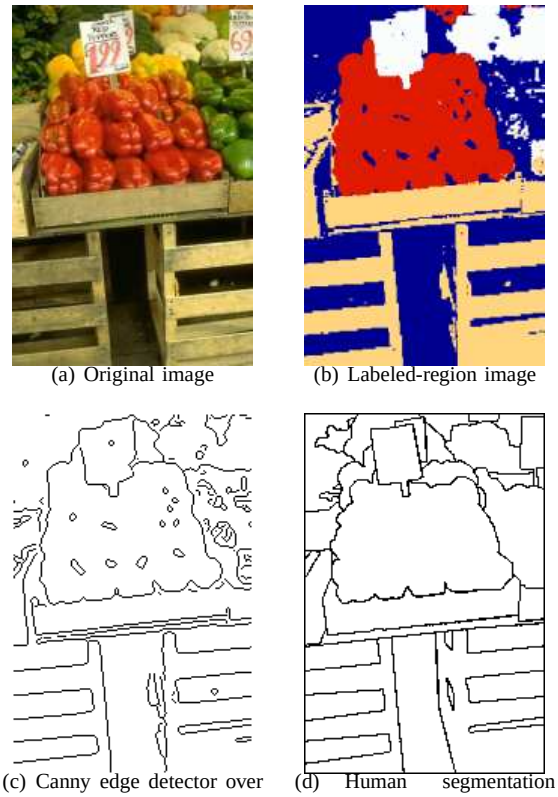


Fig. 6. Color segmentation - Computation times. Dashed line: Standard *NCut* (eigenvalue decomposition using Lanczos method). Solid line: out-of-sample extension.



(a) Original image (b) Labeled-region image

(c) Canny edge detector over the labeled-region image (d) Human segmentation taken from the Berkeley image dataset

Fig. 7. Color segmentation - "Peppers" dataset - Segmentation results. The cluster boundaries are comparable to the human segmentation

ANMMM; AWI;IWT:PhD grants, GBOU (McKnow) Soft4s, the Belgian Federal Government (Belgian Federal Science Policy Office: IUAP V-22; PODO-II (CP/01/40), the EU(FP5-Quprodis;ERNSI, Eureka 2063-Impact;Eureka 2419-FLiTE) and Contracts Research/Agreements (ISMC/IPCOS, Data4s, TML, Elia, LMS, IPCOS, Mastercard). Johan Suykens is a professor at the K.U.Leuven, Belgium. The scientific responsibility is assumed by its authors.

REFERENCES

[1] C. Alzate and J. A. K. Suykens. Kernel principal component analysis using an epsilon insensitive robust loss function. Internal report 06-03. Submitted for publication, ESAT-SISTA, K. U. Leuven, 2006.

[2] C. Alzate and J. A. K. Suykens. A weighted kernel pca formulation with out-of-sample extensions for spectral clustering methods. In *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN'06)*, pages 138–144, 2006.

[3] F. R. Bach and M. I. Jordan. Learning spectral clustering. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

[4] Y. Bengio, J. F. Paiement, P. Vincent, O. Delalleau, N. Le Roux, and M. Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

[5] P. K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE Trans. Computer-Aided Design*, 13(9):1088–1096, 1994.

[6] C. Ding and X. He. K-means clustering via principal component analysis. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 29, New York, NY, USA, 2004. ACM Press.

[7] C. Ding and X. He. Linearized cluster assignment via spectral ordering. In *ICML '04: Proceedings of the twenty-first International Conference on Machine Learning*, page 30, New York, NY, USA, 2004. ACM Press.

[8] K. Fan. On a theorem of weyl concerning eigenvalues of linear transformations. *Proc. Natl. Acad. Sci. USA*, 35(11):652–655, 1949.

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

[10] I. Fisher and J. Poland. Amplifying the block matrix structure for spectral clustering. In *Proceedings of the 14th Annual Machine Conference of Belgium and the Netherlands*, pages 21–28, 2005.

[11] M. Gu, H. Zha, C. Ding, X. He, and H. Simon. Spectral relaxation models and structure analysis for k-way graph clustering and bi-clustering. Technical report, Penn. State Univ, Computer Science and Engineering., 2001.

[12] B. Jeon, Y. Jung, and K. Hong. Image segmentation by unsupervised sparse clustering. *Pattern Recognition Letters*, 27:1650–1664, 2006.

[13] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[14] M. Meila and J. Shi. Learning segmentation by random walks. In T.K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*. MIT Press, 2001.

[15] M. Meila and J. Shi. A random walks view of spectral segmentation. In T. Jaakkola and T. Richardson, editors, *Artificial Intelligence and Statistics AISTATS*, 2001.

[16] B. Schölkopf, A. Smola, and K. R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

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

[18] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

[19] J. A. K. Suykens, T. Van Gestel, J. Vandewalle, and B. De Moor. A support vector machine formulation to PCA analysis and its kernel version. *IEEE Trans. Neural Networks*, 14(2):447–450, March 2003.

[20] S. Wang and J. M. Siskind. Image segmentation with ratio cut. *IEEE Trans. Pattern Anal. Machine Intell.*, 25(6):675–690, 2003.