Unsupervised Feature Reduction in Image Segmentation by Local Karhunen-Loève Transform

Josef Bigün
Signal Processing Laboratory
Swiss Federal Institute of Technology
Ecublens (DE) CH-1015 Lausanne Switzerland

Abstract

We propose to reduce the dimensionality of feature vectors by using the principles of Karhunen-Loève transform (KL) applied to the feature images locally and globally. The reduction is achieved by choosing the resulting basis vectors which are closest to those of the classical KL transform. An efficient implementation technique using pyramids is proposed. Experimental results are presented.

1 Introduction

The automatic segmentation of an image into regions that are homogeneous according to certain properties, such as gray level, texture or color, is an important step in machine vision, and has consequently been the topic of a an intensive research. In the general case the segmentation chain includes three steps: *Feature extraction, Feature selection, and Segmentation/Classification.* The choice for the feature extraction techniques is quite large. We refer to [2] for a review and an example of this step. Likewise, several segmentation algorithms have been developed that can segment images successfully based on multi dimensional feature spaces in an *unsupervised manner,* [10]. Depending on application, a classifier which requires training, and which is therefore *supervised,* can be used as an alternative, [7]. If in the process of image partitioning training has taken place we refer to it as supervised classification, and unsupervised segmentation otherwise.

For unsupervised segmentation purposes, the most commonly used feature reduction technique is the classical KL transform, [7]. We will sometimes refer to this as the global KL transform. For another example of unsupervised feature reduction method we refer to [12, 3]. We present a method which uses the KL transform in a local and global manner in order to perform unsupervised feature reduction for multi-class problems. The method results in an ON transform and will be referred to as the Local Karhunen-Loève, (LKL), transform. A preliminary version of the algorithm not taking into account to the computational load, and the final ordering of the coordinates with respect to class separation, was presented in [1].

In a first step, the transform is computed in a local manner, using a sliding window, so that corresponding to each pixel position an optimal local representation is obtained. These are then processed further in a second step by using the global KL transform in order to define a transform reducing the dimensionality. We test the soundness of the method by using textures from aerial images. In section 2 we describe the LKL transform. In Section 3 efficient implementation issues will be discussed. In Section 4 the experimental results will be presented. Finally in Section 5 the conclusions and a discussion will be given.

2 LKL transform

Suppose that the feature vectors $\mathbf{f} = (f_1 \cdots f_n)^t$ corresponding to the pixels in the discrete, original image are given. Within a region corresponding to one class, Ω, $\mathbf{f}(x, y)$ can be represented with very few dimensions, see Figure 1, by solving the eigen value problem:

$$S\mathbf{u}_i = \lambda_i \mathbf{u}_i \qquad \text{with} \qquad S = \sum_{(x,y)^t \in \Omega} \mathbf{f}(x,y)\mathbf{f}^t(x,y) \tag{1}$$

where $|\mathbf{u}_i| = 1$, and projecting $\mathbf{f}$ to the most significant eigen vectors. We assume that $\lambda_1 \geq \lambda_2 ... \geq \lambda_n$. If one had to choose to represent $\mathbf{f} \in \Omega$, only by using one dimension then $\mathbf{u}_1$ would be the coordinate yielding the least mean square error. Inside a region the

0-8186-2915-0/92 $3.00 © 1992 IEEE

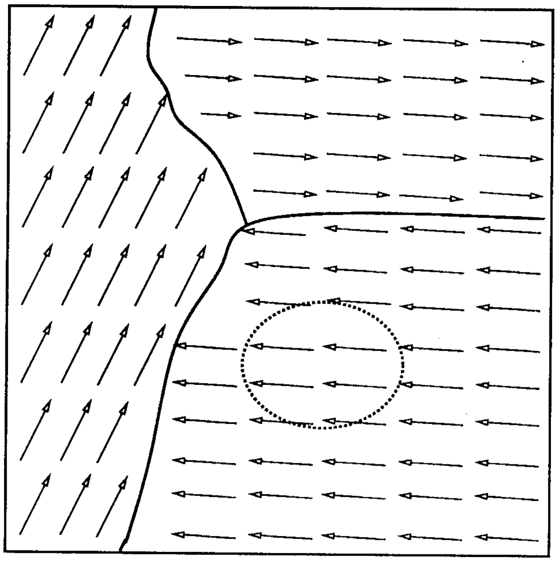


Figure 1: The area Ω, dashed, is inside a region corresponding to a class. The vectors represent the features. The vector image is spatially discrete

quantity

$$q_1 = \frac{\lambda_2 + \cdots \lambda_n}{\lambda_1 + \ldots + \lambda_n} \quad (2)$$

which is the relative average error in the truncated representation, can be expected to be small due to the slow variation of the feature vectors. However, if the region changes the optimal representation may change significantly depending on whether Ω is still within the same class or not.

In order to approximate the feature vectors in a class, we use local neighborhoods. We expect that a neighborhood whose size is chosen properly will most of the time be inside the regions corresponding to single classes as it is slid over the image. This is justified by the geometric constraint that the boundary pixels can not exceed the region pixels in number. All pixels are assumed to have class assignments. Hence, assuming that the neighborhood is within a one class region, through projections to u_1 it can be awaited that the local feature vectors as well as the other feature vectors belonging to the same class as those in the neighborhood, can be represented well. Sliding the window over all the pixels results in a *vector image* which corresponds to optimal representation directions of all neighborhoods.

Finding the principal local directions among all local directions is equivalent to the problem of local representation described above. The corresponding eigen value problem yields

$$S'\mathbf{u}'_{\mathbf{i}} = \lambda'_i \mathbf{u}'_{\mathbf{i}} \qquad \text{with} \qquad S' = \sum_{(x,y)^t \in \Omega'} \mathbf{u}(x,y)\mathbf{u}^t(x,y)$$

where Ω' represents the entire image. Thus solving this second eigen value problem yields a compromise among all candidate representations coming from the local images. The relative mean error, q'_1, can not be expected to be as small as its counter parts from the local images, q_1, since the statistics are now gathered from the entire image which in general includes areas with many different textures.

The obtained coordinates must also cause a large difference between the classes in the projected coordinates. This requirement is common with the supervised feature reduction case, see the Fisher ratio [8], in which the discrimination increases when the between class variance increases. Thus the ordering of the eigen vectors with respect to significance for class discrimination power must yet be done. In the lack of knowledge about the class statistics, we propose the ordering to be done according to the variances of the projected features. This is equivalent to the ordering of

$$v_i = \mathbf{u}'^t_i C \mathbf{u}'_i \quad (3)$$

where C is the global, trace normalized, covariance matrix. Another way to see the expression (3), which is nonnegative since C is a covariance matrix, is that the obtained eigenvectors, $\mathbf{u}_i$ are reordered according to their closeness to the global KL eigenvectors. Here we note that $\mathbf{u}'_1$, is in general not an eigenvector of C, and the eigenvectors of C constitute the global KL transform. The transformed features, $\mathbf{f}'(x,y) = (f'_1, \cdots, f'_m)^t$, after the reordering of $\mathbf{u}'_i$ according to v_i so that $v_1 \geq v_2 ... \geq v_m$, yield:

$$f'_k = \mathbf{f}^t \mathbf{u}'_{k'(k)} \qquad \text{with} \qquad k \in 1, \cdots, m. \quad (4)$$

Here m is determined by increasing m so that v_{m+1} is sufficiently small (thresholding) and $m \leq n$.

3 Implementation

Computing the LKL transform in a straight forward manner according to the functional description in the previous section can be prohibitive for its high CPU time consumption. The major cost stems from the building of the local scatter matrices, and the computation of the corresponding eigenvectors. We present some properties of the transform in order to obtain

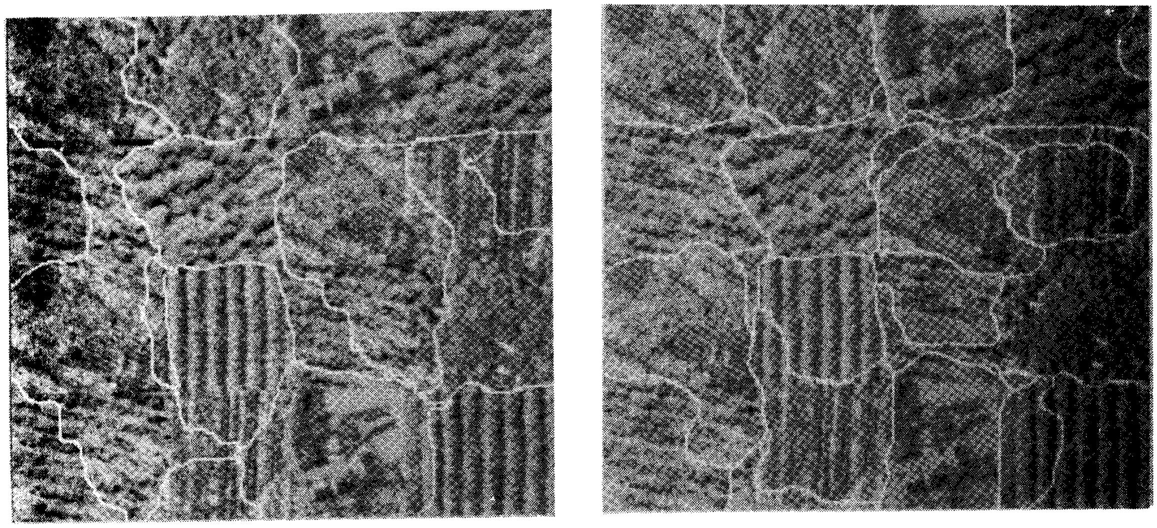


Figure 2: Feature reduction on Gabor power spectrum features using the KL, left, and LKL, right, transforms.

an efficient method for computations. First we observe the following. *Within a class the feature vectors are almost constant. Drastic changes can only occur in neighborhoods containing texture boundaries. However, the number of boundary pixels is negligible compared to the total number of points in the image.*

The principal local direction, $\mathbf{u}_1$, is the eigenvector corresponding to the largest eigen value of the local scatter matrix S:

$$S_i(x,y) = \sum_{(x,y)\in\Omega_i} w(x,y)\mathbf{f}(x,y)\mathbf{f}(x,y)^t \tag{5}$$

where Ω_i denotes the set of points belonging to the neighborhood of a pixel labeled as i, and $w(x,y)$ represents some spatially isotropic window coefficients. We note that, in the previous section, for the sake of simplicity, these coefficients were assumed to be equal to one.

An $n \times n$ symmetric matrix can be decomposed into n matrices each having the rank 1:

$$A = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^t \tag{6}$$

Here $\mathbf{u}_i$'s are eigen vectors and λ_i's are their corresponding, eigenvalues. Ideally, if S is the local scatter matrix then $1 - q_1$ should be 1 if the local constancy assumption is correct. When the local feature vectors are described by a constant perturbed with a small amount of noise, the relative error in the approximation $S = \lambda_1 u_1 u_1^t$ is still low. Thus, by using (5) and (6) the following approximation of u_1 is justified:

$$\mathbf{u}_1 = c \sum_j \sqrt{w_j}\mathbf{f}_j. \tag{7}$$

Here c is a normalization constant and equals to $|| \sum_j w_j \mathbf{f}_j ||^{-1}$ when this is non-singular, 0 otherwise. That is the local eigen vectors can be obtained by smoothing followed by a normalization. In terms of the number of arithmetic operations, this method gives a sensibly less load especially when a separable filter such as a Gaussian is utilized in the smoothing. The efficiency can be increased yet another order of magnitude if the filtering process is performed sparsely or equivalently if it is carried out by using pyramids, [4]. The actual size of the eigenvector image will be small and the construction of the global S' will result in a small computational load compared to the construction and the eigenvector computation of C. This puts the computational complexity of the method at the same order of the magnitude as the global KL transform since in both of the cases the construction of C is required. For a comparison of the results based on the direct and the normalized smoothing implementation we refer to [3].

4 Experimental results

Method

Both KL, and LKL transforms propose a complete ON coordinate system. The importance of the produced coordinate axes are ordered in accordance with a vector consisting of non-negative scalars as components, *the relevance vector.* For the KL transform this is given by the ordered eigen values while for the LKL transform it is given by the ordered global variances, linked to the coordinate vectors, (3). Normally the most relevant axes are those whose eigen values are less than a threshold. The original features are then projected to these axes. However, to facilitate our assessments we will by pass the non-trivial thresholding problem by simply selecting as many features as necessary in order to obtain the best results after the segmentation. The assessment is performed through visual inspection by taking into account the number of classes found and the boundary accuracy compared to the true borders which are known for our arranged test image.

For segmentation, we use the clustering algorithm of [11] which admits only one parameter corresponding to the minimum region size. We have used the region size of 16×16. Given the meaning of the window of the LKL transform, it was natural to use the same window size parameter as the one used for the segmentation method so that the freedom related to this parameter was removed. Moreover we used a Gaussian with $\sigma = 4.0$ as a weighing in that window.

Data and results

The test image has the size of 256×256 and consists of 16 patches illustrating different types of field and forest textures and a residential area. These patches represent 7 distinct textures which are arranged in such a way that any texture has any other as a neighbor at least once. We utilize 2 sets of features on the same test image.

The first set corresponds to a 30 dimensional feature space representing the local power spectrum. It is obtained by taking the magnitudes of the responses when a filter bank is applied to the original. The filter bank consists of the complex valued Gabor-filters, [9], with frequency responses being Gaussians whose center frequencies and bandwidths are in octave progression, [6]. Similar filter banks has been utilized in many other applications [5]. Figure 2 left illustrates the segmentation result when the KL transform was utilized. 5 classes were found using 3 transformed features (m=3). The corresponding result when using the LKL transform is given by Figure 2 right. All 7 classes were found by using 4 features.

The second set of features has 12 dimensions. It represents the local dominant orientation and the energy in the octave frequency bands of the power spectrum. The features are obtained by applying the Linear Symmetry algorithm, [2], to the Laplacian pyramid of the original. Figure 3 left with 5 classes represents the result when 4 KL transformed features were utilized in the segmentation. Figure 3 right illustrates the corresponding LKL transform results. By utilizing 7 features 6 classes were found.

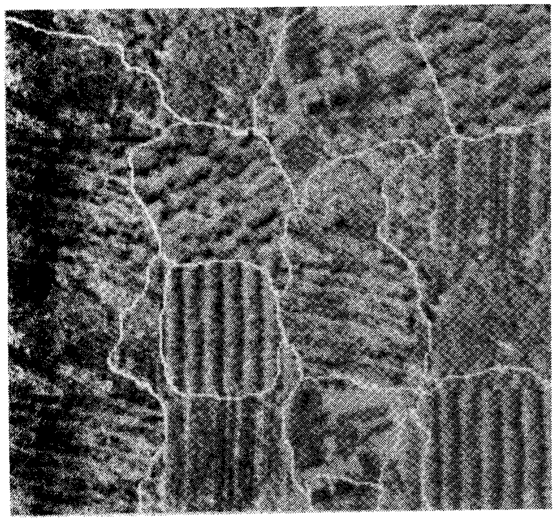
5 Discussion and conclusions

The experimental results on real texture images show that the LKL transform improves the segmentation performance. It exhibits poorer global mean square error compared to the ON system produced by KL-transform since the latter is designed to achieve that purpose. However, the region discrimination is not suffered as much as it does in the KL-transform. For image synthesis applications such as lossless transmission and coding, an optimal LMS representation can be highly important. On the contrary, judging from those results presented here and others we have performed, we conclude that one can improve the performance of the classical KL transform considerably for analysis purposes by taking into account the spatial arrangement of the features. A common limitation of both of the methods is that they are linear and metrics preserving. Thus further performance gain is possible by extending the future research to include non-linear methods. Another common problem is the dimensionality estimation once a relevance vector is obtained.

Compared to the KL transform, a local window must be defined in addition to the number of the transformed feature parameters. We have proposed to use a Gaussian because of its simplicity. While this additional a priori knowledge is in conflict with the very goal of unsupervised image segmentation, it can be seen as the price to pay in order to achieve better segmentation results.

Acknowledgment

This project is supported by Thomson CSF (France).



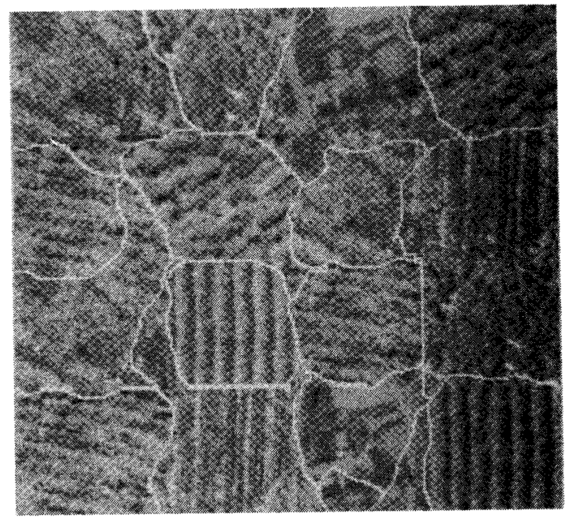


Figure 3: Feature reduction on Linear Symmetry features using the KL, left, and LKL, right, transforms.

References

[1] J. Bigün C. Horne *Feature space dimensionality reduction by using local representability* Int. conf. on signal processing, Beijing, pp. 555-558 Oct. 22-26 (1990).

[2] J. Bigün, G. H. Granlund and J. Wiklund *Multi-dimensional orientation estimation with applications to texture analysis and optical flow* IEEE-PAMI vol. 13, No. 8, pp. 775-790 (1991).

[3] J. Bigün *Unsupervised Feature Reduction in Image Segmentation by Local Transforms* Submitted to Pattern Recognition Letters, (1991).

[4] P. Burt: *Fast filter transforms for image processing* Computer graphics and image processing 16, pp. 20-51 (1981)

[5] J. G. Daugman *Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression* IEEE ASSP vol. 36, pp. 1169-1179 (1988)

[6] J. M. H. du Buf *Towards unsupervised texture segmentation using Gabor spectral decomposition* 5th Int. conf. on image analysis and processing. 20-22 september (1989), positano, Italy. pp. 65-72

[7] R. Duda and P. Hart *Pattern classification and scene analysis* J. Willey, New york, (1973)

[8] R. A. Fisher *The use of multiple measurements in taxonomic problems* Ann. Eugenics, vol. 7 pp. 179-188 (1936)

[9] D. Gabor *Theory of communication* J. Inst., Elec. Eng. vol. 93, pp. 429-459, (1946).

[10] A. K. Jain and R. C. Dubes *Algorithms for clustering data* Englewood Cliffs, NJ, (1988)

[11] M. Spann, R. Wilson *A quad-tree approach to image segmentation which combines statistical and spatial information* Pattern Recognition Vol. 18 Nos 3/4, pp 257-269 (1989)

[12] M. Unser and M. Eden *Multiresolution Feature Extraction and Selection for Texture Feature Segmentation* IEEE-PAMI, vol. 11, pp. 717-727, July (1989).