The Adaptive Subspace Map for Image Description and Image Database Retrieval

Dick de Ridder[1,2], Olaf Lemmers[1], Robert P.W. Duin[1], and Josef Kittler[2]

[1] Pattern Recognition Group
Dept. of Applied Physics, Faculty of Applied Sciences, Delft University of Technology
Lorentzweg 1, 2628 CJ Delft, The Netherlands
Telephone: +31 15 278 1845 Telefax: +31 15 278 6740
dick@ph.tn.tudelft.nl
http://www.ph.tn.tudelft.nl/~dick
[2] Centre for Vision, Speech and Signal Processing,
Dept. of Electronic & Electrical Engineering, School of EEITM,
University of Surrey
Guildford, GU2 5XH Surrey, United Kingdom

Abstract. In this paper, a mixture-of-subspaces model is proposed to describe images. Images or image patches, when translated, rotated or scaled, lie in low-dimensional subspaces of the high-dimensional space spanned by the grey values. These manifolds can locally be approximated by a linear subspace. The adaptive subspace map is a method to learn such a mixture-of-subspaces from the data. Due to its general nature, various clustering and subspace-finding algorithms can be used. If the adaptive subspace map is trained on data extracted from images, a description of the image content is obtained, which can then be used for various classification and clustering problems. Here, the method is applied to an image database retrieval problem and an object image classification problem, and is shown to give promising results.

1 Introduction

A method often used for representing image data is to use the image function on a grid of pixel positions, i.e. $I(x, y)$. A problem with this representation is that it is rather unnatural in a number of ways. In principle, although the space spanned by all pixel positions (x, y) is high-dimensional, it would seem to lend itself well for use in standard pattern recognition approaches. However, if the image is just slightly translated, rotated or scaled, the distribution of the data in the high-dimensional space changes completely, whereas to a human observer the data still looks very similar. In fact, transformed versions of an image all lie on an m-dimensional manifold in the high-dimensional space spanned by all pixel grey values, where m is the number of degrees of freedom present in the set of transformations [7]. Although this manifold may be low-dimensional, it is likely to fill a large portion of the high-dimensional space. The approach proposed here is to describe an image, or image patches, using subspaces in the high-dimensional grey value space rather than just points. A mixture of subspaces

can then be used to model the manifold discussed earlier. It draws on earlier work by Kohonen, the adaptive subspace self-organising map (ASSOM) [5] and work by Hinton et. al. [3]. Their approaches will be discussed in Sect. 2.

In Sect. 3 the basic version of the proposed method, the adaptive subspace map, or ASM, will be introduced. Next, in Sect. 4, it is applied to an image database retrieval problem. In this problem, an ASM is trained on each image. Then, a distance measure between the ASMs is used to retrieve images.

Sect. 5 discusses a slightly different problem, that of object recognition. Here ASMs are trained on a collection of images representing a certain class, and histograms of an image mapped onto these ASMs are used to classify it. Finally, Section 6 will draw some conclusions and give ideas for further work.

2 Adaptive Image Description

Kohonen [5] proposed an extension of his self-organising map, which uses subspaces $\boldsymbol{S}_j$ in each node rather than just single weights. This adaptive subspace self-organising map, or ASSOM, is based on training not just using single samples but sets E_k of slightly translated, rotated and/or scaled signal or image samples, called *episodes*. These episodes are treated as a single entity, that is, samples are assigned as a group to a subspace based on a distance measure between an episode and a subspace, which is the minimum projection error of any sample $\boldsymbol{x} \in E_k$:

$$D(E_k, \boldsymbol{S}_j) = \min_{\boldsymbol{x} \in E_k} ||\boldsymbol{x} - \hat{\boldsymbol{x}}^j|| \ , \tag{1}$$

where $\hat{\boldsymbol{x}}^j$ is the projection of $\boldsymbol{x}$ onto subspace $\boldsymbol{S}_j$. To train the ASSOM, samples drawn from a signal or an image are converted into episodes by creating slightly transformed versions of the original sample. The distance between each node and the episode is then calculated, and the winning node is defined as that node to which the episode has minimum distance. In the adaptation phase, the winning node's subspace, and that of its neighbours, is rotated to better fit the just presented episode.

The ASSOM gives good results, but is extremely slow in training. This is caused by the learning of the subspaces by rotating them, which demands careful and prudent setting of learning parameters, but also by the updating of neighbourhoods to obtain a topologically correct map. If one drops these demands, i.e. just finds the subspaces in a batch-mode operation (e.g. using principal component analysis (PCA)), and performs non-topologically correct clustering, the resulting system would be greatly simplified. Such a system would come close to the system described by Hinton et al. [3], a mixture of local linear models. However, their method does not use the idea of episodes, and was mainly used on small images containing entire objects (handwritten digits).

In this paper a system is described which combines the best aspects of these two approaches. On the one hand, knowledge of the invariances one desires the models to display can be used when creating the training set, as was done by Kohonen. On the other hand, to avoid the extremely long training times of the

ASSOM, a mixture of local subspaces is used. An overview of our proposed system will be given in Sect. 3.

3 The Adaptive Subspace Map

The basic idea of the ASM is to find a number of subspaces which describe the episodes well. This calls for both a clustering mechanism, which assigns episodes to subspaces, and a subspace-finding method, which calculates subspace parameters based on the assigned episodes. Although many algorithms could in principle be used, here we only consider the PCA algorithm for finding subspaces and a k-means like clustering method we will refer to as the k-subspaces method. The basic algorithm for the adaptive subspace method thus is:

1. Create a data set by extracting samples from one or more images and make episodes E_k by translating, rotating and/or scaling the samples
2. Assign these episodes randomly to one of the subspaces $\langle \boldsymbol{S}_j, \boldsymbol{O}_j \rangle$, $j = 1, \ldots, n$
3. Re-calculate each subspace, using a PCA on the episodes assigned to that subspace to find $\boldsymbol{S}_j$ and setting the origin $\boldsymbol{O}_j$ of the subspace to be the mean of these episodes
4. Assign each episode E_k to the closest subspace, which is that subspace $\langle \boldsymbol{S}_j, \boldsymbol{O}_j \rangle$ to which the average sample projection distance is smallest:

$$D(E_k, \langle \boldsymbol{S}_j, \boldsymbol{O}_j \rangle) = \frac{1}{|E_k|} \sum_{\boldsymbol{x} \in E_k} ||\boldsymbol{x} - \hat{\boldsymbol{x}}^j|| \ , \tag{2}$$

   where $\hat{\boldsymbol{x}}^j$ is the projection of $(\boldsymbol{x} - \boldsymbol{O}_j)$ onto $\boldsymbol{S}_j$
5. While not converged, go to 3

Note that we use the average distance of all samples as opposed to the minimum distance used by Kohonen. This stabilised convergence and did not give very different results. Also, in the ASM each subspace has its own origin $\boldsymbol{O}_j$.

Episodes are created by taking randomly shifted samples in a certain range (translation), by rotating the image over a certain range of angles and taking samples (rotation) or by scaling the image over a certain range of scales and taking samples (scaling). For the latter two, the rotation or scale range is divided into a number of equally large steps, with a small random offset.

In the experiments described in this paper, after creation of the episodes, the data is pre-processed by subtracting the average grey value from each sample and normalising its standard deviation to 1. Note that this means that the origin $\boldsymbol{O}_j$ will be zero for all subspaces.

4 Application 1: Image Database Retrieval

An ASM trained on episodes collected from images, or even a collection of images, can be seen as a descriptor of that image. This is useful for image database

applications, in which it is often a problem to define measures such that image content can be described in a compact way. A large body of literature exists dealing with indexing images based on their texture content – see e.g. Antani et al. [1] for a review. As they note, it is impossible to define a good set of features beforehand for a wide variety of images; therefore, the best approach is to be adaptive.

In our approach, the feature extraction and feature selection stages are rolled into one and performed automatically. All that remains is to define a way of using the ASMs to find distances between (classes of) images, say I^A and I^B. There are two possible strategies:

1. Train ASMs A and B on images (or classes of images) I^A and I^B and use these as a descriptor
2. Train an ASM A on image I^A (or class of images) and use the histogram of an image I^B mapped onto ASM A as a descriptor (in mapping an image, each pixel is assigned to a subspace using the window around it)

The second option seems to be preferable, as it defines distances between images I^A and I^B in terms of the content of both images and the size of the regions of similar content. However, it requires mapping each new image (or query image) onto all maps found so far, which can be a computationally intensive task. The first option is computationally much lighter, but discards information on the size of the regions responsible for each subspace in the map. For the image database application, we used the first option. The second option is found to be more applicable to classification problems. An example is discussed in Sect. 5.

4.1 Comparing ASMs

For the first approach, a distance measure between ASMs A, with n^A subspaces $\boldsymbol{S}_i^A$, and B, with n^B subspaces $\boldsymbol{S}_j^B$, can be defined:

$$D(A,B) = \max(D'(A,B), D'(B,A)) \tag{3}$$

$$D'(A,B) = \frac{1}{n^A} \sum_{i=1}^{n^A} \min_{j=1,\ldots,n^B} D''(\langle \boldsymbol{S}_i^A, \boldsymbol{O}_i^A \rangle, \langle \boldsymbol{S}_j^B, \boldsymbol{O}_j^B \rangle) \tag{4}$$

$$D''(\langle \boldsymbol{S}_i^A, \boldsymbol{O}_i^A \rangle, \langle \boldsymbol{S}_j^B, \boldsymbol{O}_j^B \rangle) = \frac{1}{d} \sum_{k=1}^{d} ||\boldsymbol{s}_k^{A_i} - \hat{\boldsymbol{s}}_k^{A_i}||^2 \ . \tag{5}$$

Here each d-dimensional subspace $\boldsymbol{S}$ is spanned by basis vectors $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_d$ and $\hat{\boldsymbol{s}}_k^{A_i}$ is the projection of basis vector k of subspace $\boldsymbol{S}_i^A$ onto subspace $\boldsymbol{S}_j^B$ (after $\boldsymbol{O}_j^B$ has been subtracted)[1]. The idea behind this measure is to find, for each

[1] Experiments were also performed in which a more principled distance measure between subspaces called the *gap* [9] was used: $D''(\boldsymbol{S}_i^A, \boldsymbol{S}_j^B) = ||\boldsymbol{P}_i^A - \boldsymbol{P}_j^B||_2$, where $\boldsymbol{P} = \boldsymbol{S}(\boldsymbol{S}^T\boldsymbol{S})^{-1}\boldsymbol{S}^T$ is the projection matrix onto subspace $\boldsymbol{S}$. However, the computational burden of this method was much higher due to the singular value decomposition needed for the calculation of the norm, and results were more or less the same.

Fig. 1. Example images from the five sequences, from left to right: news reader, queen, cathedral, hut and flag. The regions were used only in the KIDS system.

subspace in ASM A, the closest subspace in B and average this distance over all subspaces in A. The same is done for B and the maximum is taken, like in the Haussdorf distance.

As said before, the problem with this distance measure is that all information about the size of the regions responsible for the subspaces is discarded. Therefore, one can weight the distance between a subspace $\boldsymbol{S}_i^A$ and an ASM B with the relative importance of $\boldsymbol{S}_i^A$ in ASM A:

$$D'(A,B) = \frac{1}{n^A} \sum_{i=1}^{n^A} y_i^A \min_{j=1,\ldots,n^B} D''(\langle \boldsymbol{S}_i^A, \boldsymbol{O}_i^A \rangle, \langle \boldsymbol{S}_j^B, \boldsymbol{O}_j^B \rangle) \ . \tag{6}$$

It was found that

$$y_i^A = \frac{\boldsymbol{h}_i^A (1 - \bar{\boldsymbol{e}}_i^A)}{\sum_{j=1}^{n^A} \boldsymbol{h}_j^A} \tag{7}$$

gave good results, where $\boldsymbol{h}_i^A$ is the i^{th} bin of the histogram $\boldsymbol{h}^A$ of the mapping of the training image onto ASM A, and $\bar{\boldsymbol{e}}_i^A$ is the average projection error of the pixels assigned to subspace $\boldsymbol{S}_i^A$.

4.2 Experiments

A small image database was created containing images taken randomly from the MPEG7 database and stills from an hour-long video sequence of Sky news. Besides these images, five sequences of similar images of a news reader, HRH the Queen, Guildford cathedral, a hut, and a US flag were added. The goal of the experiment was, given an image in a sequence, to find the other images in that sequence. Examples of the images in these sequences are given in Fig. 1. All 200 database images, originally 24-bit colour, were converted to grey values and histogram stretched.

On each individual image, ASMs were trained for various settings of the number of subspaces n, subspace dimension d and sample window size w. To learn about the influence of the episodes, two experiments were run. In the first, for training, 200 episodes were created, each containing 5 samples translated randomly in a $[-5, 5]$ pixel range. In the second, episodes contained 15 samples: 5 translated as before, 5 rotated over a $[-45, 45]$ degree range and 5 scaled in a range of $[1.0, 1.5]$ times the original size. In both cases a round sampling window was used, with a diameter of w pixels.

Table 1. Results on the image database retrieval problem. Numbers shown are ranks of the images sought in the retrieval order.

Sequence	Translation invariance ASMs		All-invariance ASMs		KIDS
	2D subspaces, $n = 20, w = 12$	4D subspaces, $n = 8, w = 16$	2D subspaces, $n = 20, w = 8$	4D subspaces, $n = 12, w = 8$	
news reader	1,6,7,18	1,2,3,7	1,2,4,12	1,2,3,4	17,19,20,23,25
queen	4,10,16,20	4,6,16,22	1,5,20,21	4,6,20,36	14,15,17,21
cathedral	55,124,184,198	110,122,179,193	126,192,194,198	110,186,197,198	2,4,5,6
hut	2,5,16,50	1,2,3,40	7,9,12,57	1,2,3,83	1,2,3,5
flag	1,2,4,5,10	11,28,29,44,73	1,2,3,4,13	1,2,3,8,34	24,25,30,35,50
cathedral (2)	12,50,59,92	16,17,25,108	1,22,50,111	4,5,23,97	-

After training, for each sequence r one image I_r^{tst} was used as a test image. The distance $D(A_r^{tst}, A_i), i = 1, \ldots, 199$ between the ASM A_r^{tst} trained on this image and all other ASMs was calculated using equations 3-7 and the images were ordered by the distance of their ASM to that of the test image. Finally, the ranks of the other images in sequence r, $I_{r,i}^{trn}$ were noted. Table 1 shows the results for various settings of parameters.

As a comparison, the same queries were performed using a state-of-the-art system called KIDS [8]. This system can also handle colour features, but for the comparison only texture features (DCT, Gabor and wavelet) were used. KIDS requires the user to specify regions to search for in the database; the regions used are shown in Fig. 1. A threshold of 0.5 was used which gave optimal overall results (for more information, see [8]). The results are also given in Table 1.

The table shows that the ASM method gives promising results, even compared to KIDS. For most of the query images the other images in the sequence are ranked high. Also, training on episodes with translated, rotated and scaled samples pays off for most sequences. Notable exceptions are the cathedral and hut sequences, which are the only two sequences consisting of high-resolution images, with areas of fine texture. The ASMs code this texture quite precisely, so that different views of the cathedral give quite different ASMs. If the cathedral experiment is repeated on reduced versions of the images (by 50% in both the x and y direction), losing the high-frequency textures, results are much better; see the cathedral (2) row in Table 1. Interestingly, the two problem queries are where KIDS performs best, indicating that the two techniques might be complementary.

Of course, it is possible to use a large number of settings for the number n of subspaces to use and the sampling window size w. Unfortunately, space does not permit giving these results. For most other settings the overall results were worse, although on individual sequences they would sometimes be slightly better.

5 Application 2: Object Image Classification

The ASM can also be used as an adaptive description of a class of images. The histograms of an image mapped onto a class ASM can then be used for classification. This idea has been explored earlier by e.g. Idris and Panchanathan [4],

who use vector quantisation on an entire set of images and use histograms of images mapped onto the code book vectors as descriptors. Another example is the work of Lampinen and Oja [6], in which clusters are found in a space spanned by Gabor filters at different resolutions, and a supervised layer is applied for classification.

5.1 Experiments

For our experiments, a small data set of images of 6 different classes (book cases, chess pieces, mugs, workstations, a tea flask and some bridges) was created. All images were acquired using a Sony digital camera, re-sized to 320×240 pixels, converted to grey values and histogram stretched. Per class, 9 training images and 6 test images were used. The intra-class variation between objects was quite high, since objects were photographed at different distances and against different backgrounds. Also, the inter-class distance was kept low for the object images (chess pieces, mugs and the tea flask) by taking photographs of each of these against three different backgrounds. For some examples, see Fig. 2.

An ASM A_c with n subspaces of d dimensions each was trained for each class $c = 1, \ldots, 6$, on 900 episodes taken from 9 training images I_c^{trn}, using a round window with a diameter of w pixels. Episodes contained 5 samples translated randomly in a $[-5, 5]$ pixel range. After training, for each class c all training images $I_{c,i}^{trn}$ were mapped onto their ASM A_c, and an n-bin class histogram $\boldsymbol{h}_{c,i}^{trn}$ was created by counting the relative number of pixels assigned to each subspace (i.e. the number of pixels assigned to a subspace divided by the total number of pixels in the image). The mean $\boldsymbol{\mu}_c$ and covariance matrix $\boldsymbol{C}_c$ of these histograms were then used as class descriptions.

Each test image $I_{k,j}^{tst}$ (class $k = 1, \ldots, 6$; image $j = 1, \ldots, 6$) was mapped onto each of the class ASMs A_c. The histograms $\boldsymbol{h}_{k,j,c}^{tst}$ of these mappings were then used to calculate the Mahalanobis distance to each of the classes:

$$D_M(I_{k,j}^{tst}, A_c) = (\boldsymbol{h}_{k,j,c}^{tst} - \boldsymbol{\mu}_c)^T \boldsymbol{C}_c^{-1} (\boldsymbol{h}_{k,j,c}^{tst} - \boldsymbol{\mu}_c) \ . \tag{8}$$

Due to the small number of training images, some regularisation was necessary: $\boldsymbol{C}_c = \boldsymbol{C}_c + 10^{-4}\,\boldsymbol{I}$. Each test image $I_{k,j}^{tst}$ was then assigned to that class c which gave the lowest Mahalanobis distance:

$$c = \arg\min_c D_M(I_{k,j}^{tst}, A_c) \ . \tag{9}$$

The results of these experiments, performed for a range of $n = 8$, 12, 16 or 20 subspaces, subspace dimensions $d = 2$ or 4, and a sample window size of $w = 8$ or 16, are shown in Fig. 3. The best result obtained is a test error of 11% (4 out of 36 images classified incorrectly), which is quite reasonable given the difficulty of the problem. The window size does not seem to be too important, as for both $w = 8$ and $w = 16$ the optimum is reached. There is an optimal number of subspaces, but again the exact choice seems not too critical. Furthermore, 2-dimensional subspaces seem to be sufficient; 4-dimensional subspaces lack the distinctiveness needed to describe class-specific image information well.

Fig. 2. Some training images in the object image data set: book cases, workstations, bridges, chess pieces (3×), mugs (3×) and tea flasks (3×).

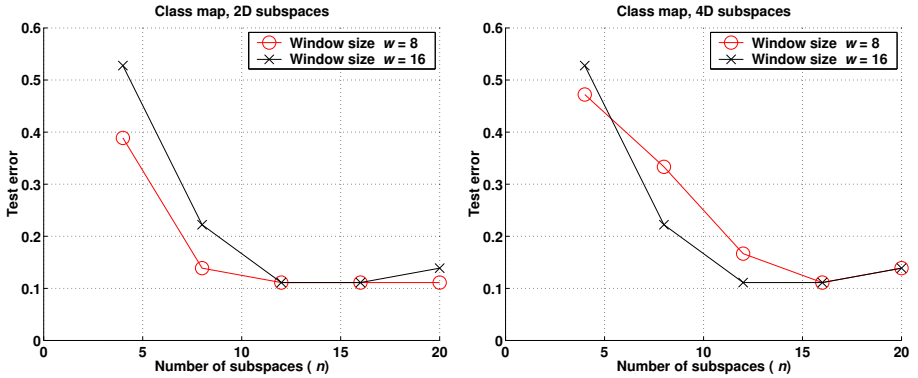

Fig. 3. Test error for various settings of the number of subspaces n, the sample window size w and subspace dimensionalities $d = 2$ (left) and $d = 4$ (right).

Fig. 4. The four incorrectly classified images.

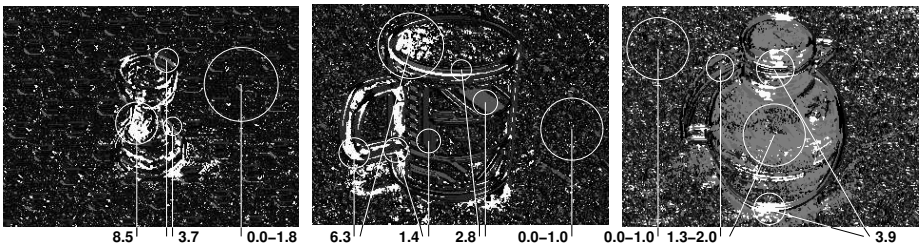

Fig. 5. Feature ranking for an image containing a chess piece (left), a mug (middle) and a tea flask (right). The numbers indicate the ranks for some regions.

It is interesting to look at the misclassifications of the best performing ASMs. Mostly the same test images are misclassified for a variety of settings: three images in the mugs class and one workstation image. These are shown in Fig. 4. The three images of mugs are the only three images (in both train and test set) in which the ears of the mug are not visible. These images are classified as chess pieces. The workstations image is the only one in which a row of books is also visible, and is labelled as a book case. In all these cases, the difference in Mahalanobis distance between the true class and the class the image was labelled as, was small.

5.2 Feature extraction

To investigate what features are found by the ASMs, and whether the ASMs do not merely describe the background, the subspaces were ranked as follows. Only the classes of images containing chess pieces, mugs and teaflasks were considered, as these shared the same backgrounds. Each subspace $\boldsymbol{S}_i^{A_c}$, $i = 1, \ldots, n$ representing class c was ranked by calculating the average Mahalanobis distance of the training images of the other classes to that class c, using only mapping histogram bin $\boldsymbol{h}_i$. This distance was then used to label the image: the brighter the colour, the better the feature. Figure 5 shows three examples of these rankings. The background is mapped on just a few subspaces, more or less randomly, but some informative features have been found for each of the classes: curved edges and uniform regions for the chess pieces, the curved ears of the mugs and curves and large uniform regions for the tea flask. Of course, it is the Mahalanobis distance that makes use of these features.

6 Conclusions

An adaptive image description method was presented, which uses subspaces to describe translated, rotated and/or scaled versions of patches extracted from images. The resulting description (ASM) can be used to segment images [2], but in this paper the focus was on using it for classification purposes.

First, given the distance measure between ASMs introduced in Sect. 4, ASMs can be used to compare images. This method was shown to be applicable to image database retrieval problems. Although the database used was small, the results were quite good compared to a state-of-the-art system, given the fact that only texture and edge information is used (since the average grey value is removed from the samples).

Second, mappings of images on ASMs can be used. This was demonstrated on an object recognition problem. Histograms of images mapped onto class-specific ASMs indicate the size of regions present in an image containing a specific texture or structure. These histograms can then be used to classify images. Experiments show that significant features are found and recognised.

In the experiments performed some important settings, such as the number of subspaces to use within each ASM, subspace dimensionality and sampling window size, were optimised by hand. For easier applicability, it would

be interesting to investigate optimising these automatically, or perhaps to use a multi-scale approach. Another possible extension is the combination of the technique with other feature extraction mechanisms, the most important of which would be the use of colour information. Finally, we intend to investigate other subspace-finding algorithms, such as independent component analysis (ICA) or even non-linear methods such as multi-dimensional scaling (MDS), which might be more applicable to some problems.

Acknowledgements

This work was partly supported by the Foundation for Computer Science Research in the Netherlands (SION), the Dutch Organisation for Scientific Research (NWO) and the Engineering and Physical Sciences Research Council (EPSRC) in the UK (grant numbers GR/M90665 and GR/L61095). The first author would like to thank the Centre for Vision, Speech and Signal Processing and the EPSRC for allowing him to visit the centre as a visiting fellow for six months. Finally we would like to thank Kieron Messer for helping us in comparing the ASM to KIDS.

References

[1] S. Antani, R. Kasturi, and R. Jain. Pattern recognition methods in image and video databases: past, present and future. In A. Amin, D. Dori, P. Pudil, and H. Freeman, editors, *Advances in Pattern Recognition, Proceedings of SPR'98 and SSPR'98*, pages 31–53, Berlin, 1998. IAPR, Springer-Verlag.

[2] D. de Ridder, J. Kittler, O. Lemmers, and R.P.W. Duin. The adaptive subspace map for texture segmentation. In *Proceedings of the* 15^{th} *IAPR International Conference on Pattern Recognition*, 2000. To appear.

[3] G.E. Hinton, P. Dayan, and M. Revow. Modelling the manifolds of images of handwritten digits. *IEEE Transactions on Neural Networks*, 8(1):65–74, 1997.

[4] F. Idris and S. Panchanathan. Image indexing using vector quantization. In W. Niblack and R.C. Jain, editors, *Storage and retrieval for image and video databases III*, volume 2420 of *Proceedings of SPIE*, Bellingham, WA, 1995. SPIE, SPIE.

[5] T. Kohonen, S. Kaski, and H. Lappalainen. Self-organized formation of various invariant-feature filters in the Adaptive-Subspace SOM. *Neural Computation*, 9(6):1321–1344, 1997.

[6] J. Lampinen and E. Oja. Distortion tolerant pattern recognition based on self-organizing feature extraction. *IEEE Transactions on Neural Networks*, 6(3):539–547, 1995.

[7] H. Lu, Y. Fainman, and R. Hecht-Nielsen. Image manifolds. In N.M. Nasrabadi and A.K. Katsaggelos, editors, *Applications of Artificial Neural Networks in Image Processing III, Proceedings of SPIE*, volume 3307 of *SPIE Proceedings Series*, pages 52–63, Bellingham, WA, 1998. SPIE, SPIE.

[8] K. Messer and J. Kittler. A region-based image database system using colour and texture. *Pattern Recognition Letters*, 20(11-13):1323–1330, 1999.

[9] G.W. Stewart. Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *SIAM review*, 15:727–764, 1973.