Multiple Eigenspace Models for Scene Segmentation and Occlusion Removal

Arnab Dhua[1], Florin Cutzu[1], Durgesh Dewoolkar[1], and Stephen Kiselewich[2]

[1] Indiana University, Bloomington IN 47405, USA,
adhua@cs.indiana.edu
[2] Delphi Corporation, Kokomo, IN 46904, USA

Abstract. We present a method that uses eigenspace models to segment an input image into a foreground and a background component. The algorithm segments the input image into the two components, removes the mutual occlusions among the objects of the foreground and background component, and reconstructs the occluded portions of both the foreground and the background component. The problem is formulated as a nonlinear optimization and an approximate solution is found by an iterative process that alternates between input image segmentation and component reconstruction, gradually improving the two components extracted from the input image. The novelty of this approach lies in the use of multiple eigenspaces to achieve a model-based segmentation of the input image in an iterative framework. This method yields segments that correspond to meaningful real-world objects even in the presence of occlusions, and these segments can be directly used for other tasks like object recognition. This method differs from the traditional segmentation algorithms as it is not obtained in the usual bottom-up manner but is model-guided. We demonstrate the utility of the algorithm in the segmentation and recognition of partially occluded humans in an office environment.

1 Introduction and Background

We introduce a model-based approach to image segmentation and occlusion removal. Our method requires models (currently, eigenspaces) for the objects of interest (the foreground image component) as well as for the space of backgrounds. The algorithm labels the pixels of the input image as background or foreground. In addition, the algorithm reconstructs the occluded parts of both the foreground and the background components. The reconstructed foreground component can then be passed on to an object recognition module.

Traditionally, image segmentation has been based on bottom-up, model-free methods. Broadly speaking, segmentation techniques [1, 2] are contour-based, which work by linking image edges into smooth closed contours, or region-based, which seek an optimal partitioning of the image into regions of homogeneous texture or color. However, segmenting an image into homogeneous regions is not equivalent to segmenting the individual objects in the scene, for at least two reasons. First, an object can be composed of several homogeneous regions. Second, objects of interest can be occluded. The grouping of the homogeneous object sub-regions or the reconstruction of the occluded object parts is possible if prior knowledge—a model—of the class of objects of interest is available. Our method uses eigenmodels to segment and reconstruct the objects of interest, even if they are composed of regions with different textures.

The eigenface method [3] forms the basis of numerous appearance-based object recognition schemes, for example [4]. Unfortunately, the eigenspace method breaks down in the presence of occlusions. One of the contributions of the present work is a new method for handling occlusions. A method for dealing with occlusions in eigenspace-based object recognition was described in [5]. Instead of determining the eigenspace coefficients by projecting the entire input image onto the model eigenspace, the authors project a subset of the image pixels, thus achieving robustness to occlusion. The resulting reconstructed image is compared with the input image. If the two images are close enough, and if the number of image pixels giving rise to the coefficients is large enough, an acceptable hypothesis is said to have been formulated. A set of such hypotheses is generated using different sets of pixels. Competing hypotheses are then subject to a selection procedure based on the Minimum Description Length (MDL) principle. The authors' experiments indicate that their approach can reject outliers (noise) as well as deal with occlusions. Leonardis and Bischof [6] proposed a constrained search method which assumes that the eigenspace coordinates of the models are discrete points or lie on a parametric manifold. Thus, the process of generating the hypotheses in [5] was replaced by search for the closest training point in the eigenspace (or the parametric manifold). Bischof and Leonardis [7] showed that their method can be applied to convolved and sub-sampled images yielding the same value of coefficients. This allowed for an efficient multi-resolution approach, where the values of the coefficients can be propagated through scales. In [8] Hadjidemetriou and Nayar propose improvements to [5]. The authors derive criteria for selecting subsets of image pixels that maximize the recognition rate. The method is based on an analysis of sensitivity of the subspace to image noise. The authors present a window selection algorithm as well as a pixel selection algorithm. Paulus et al. have extended the work of Leonardis and Bischof and put it to practical use [9]. They propose that the random selection scheme in [5] can be improved by incorporating additional knowledge about object properties e.g., local texture, color, average intensity. The implementation has been tested on typical objects from office environments and also on objects commonly found in hospitals. A different approach to the problem of occlusions is given in [10]. The authors propose a view-based recognition method based on an eigenspace approximation to the Hausdorff measure. The authors address the problem of occlusions and clutter by matching intensity edges robustly via the Hausdorff measure, rather than directly comparing the views themselves. This combination of eigenspaces and the Hausdorff measure yields a system that has both the speed of subspace methods and the robustness of the Hausdorff measure.

2 The Proposed Method: Image Segmentation and Occluder Removal using Multiple Eigenspaces

We represent a scene by two eigenspaces, each corresponding to a subset of objects in the scene. The two sets of objects making up the scene occlude each other: one can think of them as constituting the "foreground" and "background" of the image, although we allow the background to occlude the foreground as well. The goal of the algorithm is, given an image of the scene, to remove the occlusions and separate the two sets of objects.

Let these eigenspaces be $\boldsymbol{F}$ (foreground) and $\boldsymbol{B}$ (background). The input image $\boldsymbol{x}$ is modeled as consisting of two mutually occluding components, a foreground component $\boldsymbol{f} \in \boldsymbol{F}$ and a background component $\boldsymbol{b} \in \boldsymbol{B}$. These two components are combined using a binary image-sized mask $\boldsymbol{m}$ that allows only one of the two components to be visible at any pixel location. If at a pixel location the mask is one then only the foreground component is visible and if the mask is zero then only the background component is visible. Thus:

$$\boldsymbol{x} = \boldsymbol{M}\boldsymbol{f} + (\boldsymbol{I} - \boldsymbol{M})\boldsymbol{b} \tag{1}$$

where $\boldsymbol{M}$ is a diagonal matrix with the vector $\boldsymbol{m}$ on the diagonal and $\boldsymbol{I}$ is the unit matrix. The goal of this paper is the estimation of $\boldsymbol{f}$, $\boldsymbol{b}$ and $\boldsymbol{m}$ using x, $\boldsymbol{F}$, $\boldsymbol{B}$ and certain smoothness assumptions about the mask $\boldsymbol{m}$. If ϕ, β are the unknown coordinate vectors of $\boldsymbol{f}$ and $\boldsymbol{b}$ in the spaces $\boldsymbol{F}$ and $\boldsymbol{B}$, and the (known) means corresponding to the two spaces, respectively, $\bar{\boldsymbol{f}}$ and $\bar{\boldsymbol{b}}$, then the relation above becomes:

$$\boldsymbol{x} = \boldsymbol{M}(\boldsymbol{F}\phi + \bar{\boldsymbol{f}}) + (\boldsymbol{I} - \boldsymbol{M})(\boldsymbol{B}\beta + \bar{\boldsymbol{b}}) \tag{2}$$

Our problem reduces to estimating ϕ, β and $\boldsymbol{M}$ given $\boldsymbol{x}$, $\boldsymbol{F}$, $\boldsymbol{B}$, and $\bar{\boldsymbol{f}}$ and $\bar{\boldsymbol{b}}$. The problem is ill-posed if the spaces $\boldsymbol{F}$ and $\boldsymbol{B}$ are linearly dependent. However, linear independence is not necessary. The degree of linear dependency of two spaces is measured by the angle they form, an angle of 90° corresponding to linear independence and an angle of 0° to linear dependence. For the problem to be solvable, the angle should not be very close to 0°.

Separation algorithm Solving for the image-sized vectors $\boldsymbol{f}$ and $\boldsymbol{b}$ requires determining the coefficient vectors ϕ and β as well as the image-sized occlusion mask $\boldsymbol{m}$. Typically, under the eigenspace model, the coefficient vectors ϕ, β are much smaller than $\boldsymbol{f}$ and $\boldsymbol{b}$. However, the problem remains under-determined.

To render the problem well-posed, we imposed a smoothness constraint on the mask m. We assumed that the binary occlusion mask m is "smooth": its neighboring pixels tend to be similar. This constraint is used in Step 2 of the algorithm below.

The steps of the algorithm are as follows:

1. Reconstruct the input image $\boldsymbol{x}$ using the space $\boldsymbol{F}$ obtaining an initial estimate $\hat{\boldsymbol{f}}$ of the foreground component, and the space $\boldsymbol{B}$ obtaining an initial estimate $\hat{\boldsymbol{b}}$ of the background component:

$$\hat{\boldsymbol{f}} = \boldsymbol{F}\boldsymbol{F}^T(\boldsymbol{x} - \bar{\boldsymbol{f}}) + \bar{\boldsymbol{f}} \;\text{ and }\; \hat{\boldsymbol{b}} = \boldsymbol{B}\boldsymbol{B}^T(\boldsymbol{x} - \bar{\boldsymbol{b}}) + \bar{\boldsymbol{b}} \;.$$

2. Obtain an estimate of the mask $\hat{\boldsymbol{m}}$ (and therefore $\hat{\boldsymbol{M}}$) by comparing locally the original image $\boldsymbol{x}$ with the estimated component images $\hat{\boldsymbol{f}}$ and $\hat{\boldsymbol{b}}$. Roughly speaking, subject to smoothness constraints, if at pixel i the input image $\boldsymbol{x}$ is more similar to $\hat{\boldsymbol{f}}$ than to $\hat{\boldsymbol{b}}$ then $\hat{\boldsymbol{m}}(i) = 1$, otherwise $\hat{\boldsymbol{m}}(i) = 0$. The computation of the mask is carried out using an efficient graph algorithm detailed below.
3. Use the estimated mask $\hat{\boldsymbol{M}}$ to obtain an improved estimate of the foreground and background components. This is achieved by finding the eigenspace coordinate vectors $\hat{\phi}$ and $\hat{\beta}$ that minimize the image reconstruction error:

$$(\hat{\phi}, \hat{\beta}) = \arg\min_{\phi,\beta} \|\boldsymbol{x} - \hat{\boldsymbol{M}}(\boldsymbol{F}\phi + \bar{\boldsymbol{f}}) - (\boldsymbol{I} - \hat{\boldsymbol{M}})(\boldsymbol{B}\beta + \bar{\boldsymbol{b}})\|_1 \tag{3}$$

Note that we use the L_1 norm above, as it gives much better results than the L_2 norm in practice. Due to the use of the 1-norm, the minimization must be carried out numerically. Then assign:

$$\hat{\boldsymbol{f}} = \boldsymbol{F}\hat{\phi} + \bar{\boldsymbol{f}} \quad \text{and} \quad \hat{\boldsymbol{b}} = \boldsymbol{B}\hat{\beta} + \bar{\boldsymbol{b}} \ . \tag{4}$$

4. Go to Step 2, or stop when changes in $\hat{f}, \hat{b}$ are small.
5. Return the estimates $\hat{\boldsymbol{f}}, \hat{\boldsymbol{b}}, \hat{\boldsymbol{m}}$.

The Mask as a Minimal Cut in a Graph The computation of the mask in Step 2 of the separation algorithm above is achieved by finding the minimum cut in a graph [11]. The input image x is modeled as an undirected graph: the pixels are nodes, and only adjacent pixels are connected by graph edges. The cost of a graph edge is the similarity s of the two pixels. The similarity of pixels i and j is $s(i,j) = 1 - |g(i) - g(j)|$ where g is image intensity normalized to a range between zero and one. Thus, cuts between similar pixels are discouraged. This implements the smoothness constraint on the mask.

There are two additional nodes that are connected to all the pixel nodes: a source node s and a sink node t. The source node corresponds to the foreground component, and the sink node corresponds to the background component. The cost of the edge linking a pixel to s is the similarity of the pixel to the corresponding pixel in the foreground component. The cost of the edge linking a pixel to t is the similarity of the pixel to the corresponding pixel in the background component. The minimal cut in this graph separates the source from the sink.

As a result of the cut, the nodes (image pixels) will be separated into two disjoint groups: a group of foreground (source) pixels and a group of background (sink) pixels. Pixels in the foreground group will tend to be similar to the pixels in the foreground component; pixels in the background group will tend to be similar to the pixels in the background component; and the boundaries between the foreground and background pixel groups in the input image will tend to conform to the edges (contours) of the input image, since cutting of links between adjacent dissimilar pixels is encouraged. The mask $\boldsymbol{m}$ is derived from the minimal cut by setting the $\boldsymbol{m}(i) = 1$ for the pixels i in the subset of nodes linked to the source and $\boldsymbol{m}(i) = 0$ for the pixels in the subset of nodes linked to the sink.

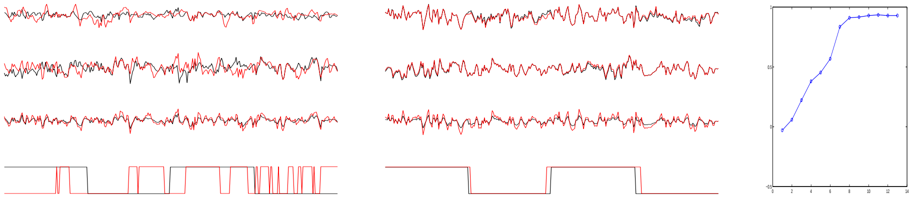

Fig. 1. Left: first iteration. From top to bottom: foreground component, background component, mask. Black: true signals. Red: estimated signals. Middle: final iteration. Same curves as on the left. The correlation between the true components and the reconstructions exceeds 0.98. The angle between the foreground and background eigenspaces is 34°. Right: correlation between the estimated mask and the true mask as a function of number of iterations.

Note that this segmentation procedure does not depend directly on the texture or grayscale values at any region in the image, rather it depends on the similarity of the pixel location to either the foreground component or the background component.

3 Numerical experiments

One-dimensional signals The algorithm was first tested on 1-D signals. An image displaying a collage of face images was used as source of data. Half of the pixel rows of the image, randomly selected, were used to generate the foreground eigenspace $\boldsymbol{F}$; the remaining rows generated the background eigenspace $\boldsymbol{B}$. The foreground and the background components were generated by taking random linear combinations of these two bases (and adding the means). Depending on how the image rows were assigned to the two spaces, the degree of linear dependency between $\boldsymbol{F}$ and $\boldsymbol{B}$ varied (row correlation decays with distance). This dependency was measured by the angle between the two spaces: a small angle indicates that the two spaces are nearly linearly dependent. We conducted several experiments, at different degrees of linear dependency between $\boldsymbol{F}$ and $\boldsymbol{B}$. A typical result is displayed in Fig. 1, demonstrating good reconstruction despite a small angle between spaces $\boldsymbol{F}$ and $\boldsymbol{B}$. In all our experiments, the correlation between true and reconstructed signals exceeded 0.95. The mask was almost always perfectly reconstructed, while the estimated foreground and the background components displayed imperfections in the regions where they were occluded in the observed signal. We observed that smaller the angle between the foreground eigenspace and the background eigenspace, slower the convergence of the algorithm.

Texture separation In a second set of experiments, we used the Vision Texture (downloaded from www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html) texture database to generate foreground and background eigenspaces. Each type of texture was used to derive an eigenspace. In one experiment, the two eigenspaces

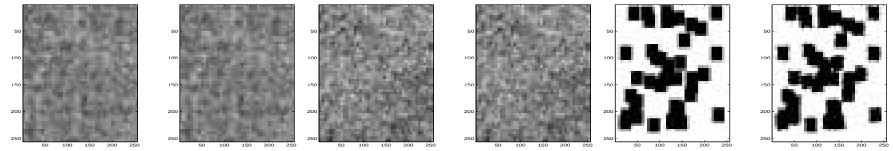

Fig. 2. Left to right: 1. True foreground component. 2. Reconstructed foreground component. 3. True background component. 4. Reconstructed background component. 5. True mask. 6. Reconstructed mask.

were derived from, respectively, rock and fabric texture images. All images were 256 × 256 pixels. The fabric eigenspace had 79 components, and the rock eigenspace, 99. The angle between the two eigenspaces was 85°. The random mask had 50% of its pixels opaque (zero). The result, shown in Fig. 2, was typical: the foreground and background components were always nearly perfectly recovered (the correlations to the true components exceeded 0.95 for all textures), but the reconstruction of the mask was noisy.

4 Segmenting and De-occluding People in an Office Environment

The algorithm was tested on a difficult real-world task: the segmentation and extraction of humans in an office environment. The segmented images were used for the recognition of the human subjects in a recognition experiment.

The foreground image set, from which the foreground eigenspace was derived, consisted of images of two people (a male and a female). The subjects were photographed frontally, from approximately the same distance, standing against a patterned (flowers and leaves) fabric curtain. In the images, the human subjects assumed various poses—arms parallel to the the body, arms folded, arms held behind the head, hands on the head, hands on waist, etc. The male subject was taller than the female, and since all images were taken from roughly the same distance, their image sizes were different as well. A degree of translational variability was also introduced, by the subjects being shifted to either side of the image center in different images, by approximately half the body width of the subjects. As a result of these manipulations, the foreground image set exhibited substantial variability. Examples of images used for the construction of the foreground eigenspace are shown in Fig. 3.

All the images were taken under similar illumination conditions, i.e., under florescent lights. Image size was 460 × 245 pixels. 50 images of the male subject and 50 images of the female subject were used to derive a single foreground eigenspace. We retained 90% of the variance in the eigenspace, thus 41 eigenvectors were used in the foreground eigenspace in our experiments.

The background image set was collected at approximately the same scale in an office environment that included a mobile chair; the chair served as the occluder of the human subject in the test images. These images were taken with slight lateral camera shifts and with the position of the chair being varied by small translations and rotations. The chair was rotated about its axis by approximately ±50°. The chair was also shifted left and right from an initial position by approximately half of its width. Examples of images used for the construction of the background eigenspace are shown in Fig. 4. Again, all the images were taken under similar illumination conditions, i.e., under florescent lights. As before, all the images were 460×245 pixels. 52 images were used for computing the background eigenspace. We retained 90% of the variance in the eigenspace, thus 14 eigenvectors were used in the background eigenspace.

The angle between the two eigenspaces was 88.28° indicating that globally (but not necessarily locally), their linear dependency is very low.

The test images were taken in the same office environment used in deriving the background eigenspace, from approximately the same viewpoint as in the training stage, but with the human subject present and the chair partially occluding the human, and the human occluding the rest of the office scene background. Again, all the images were taken under similar illumination conditions, i.e., under florescent lights. As before, all the images were of size 460×245. 18 images, 9 of the male subject and 9 of the female subject were used for testing the performance of the segmentation. The clothing worn by the subjects was the same as during the training phase. The poses of the human subject in the test images were roughly the same as the poses in the foreground training phase.

Segmentation and de-occlusion The algorithm extracted the foreground object (the human) from the background and de-occluded the human as well as the background: removed the chair and showed the human behind it and removed the human and revealed the background behind him/her.

We obtained good segmentation results for all images in the test set. We found that the segmentation was not dependent on the pose of the person, so long as the test image has the person in a pose similar to one of the poses present in the training set. The only defect in segmentation observed in our test set of 18 images was the incorrect inclusion of the right arm of the male subject into the background component. This can be accounted to the fact that this particular pose had only one example in the image set used to construct the foreground eigenspace. This defect was eliminated when we used an eigenspace with the entire variance retained. On the whole, the segmentation results were better for the male subject. The male subject had uniform clothing without much variation in texture and gray-level values. The clothing of the female subject had significant variation in texture, and hence the images of the female subject were more strongly affected by 3-D translation. This accounts for the slightly more noisy segmentation results, especially below the waist, for the female subject, as can be seen in Fig. 6.

Typical segmentation results are shown in Figs. 5 and 6. The algorithm terminated when the change (decrease) in the error function dropped below a small threshold. Plots of the dependence of the error function (3) used for the minimization vs. the iteration index are shown on the right in Figs. 5 and 6: their monotonic decrease indicates convergence.

Recognition results Next, we tackled the problem of recognizing the human subject in the input image. Since our goal was to investigate the utility of the separation algorithm as a preparatory stage to recognition, we simply used nearest neighbor as recognition method. Given a test image, its nearest neighbors in the foreground and background training sets were determined. This provided the baseline recognition rate. Then, the nearest neighbor in the foreground training set of the foreground component extracted from the input image was determined. Similarly, the nearest neighbor in the background training set of the background component of the input image was determined.

A correct match in the foreground set corresponds to the retrieval of the same subject (male or female) in a similar pose as in the input image. A correct match in the background set corresponds to the retrieval of a similar position of the chair and a similar camera angle as in the input image. The nearest neighbor match in the foreground image set using the original input image obtained the wrong person in nine out of the 18 test images and obtained the wrong pose in 12 out of the 18 test images. The nearest neighbor match in foreground image set using the foreground component extracted from the input image obtained the wrong person in just one out of the 18 test images and obtained a wrong pose in only six out of the 18 test images. The nearest neighbor match in the background image set using the original input image obtained a completely wrong position of the chair in five out of the 18 test images. The matches using the background component of the input image obtained a wrong position of the chair in only two of the 18 test images. The matches using the segmented input image were better (the position of the chair was more similar) than the ones using the original input image in 11 out of the 18 test images. In Fig. 7 the nearest neighbor match in the foreground set (the second image), of the original image is incorrect. When the foreground component extracted from the image is used for matching a better match is obtained (fourth image). In this example, the background match is good using either the original or the segmented image. Similarly, in Fig. 8 the nearest neighbor match in the background set using the original image is not as good as the closest match to the background component of the image. In this example the foreground match is correct using either the original or the segmented image. It is interesting to note how the nearest neighbor matches improve over the iterations. Fig. 9 shows the nearest neighbor match obtained during successive iterations. The figure displays, from left to right the input image and the nearest neighbor matches obtained with: the segmented image after the first, second, third, fourth and fifth iterations respectively. The match gradually improves in quality through the iterations. From the fifth iteration onwards the nearest neighbor match does not change for the example presented.

5 Discussion

Our algorithm uses information provided by the background and foreground eigenspaces to segment the input image. To obtain a correct solution, the foreground and background eigenspaces must have a low degree of linear dependency,

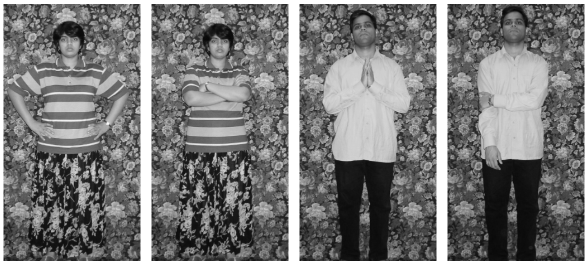

Fig. 3. Some of the training images used to generate the foreground eigenspace.

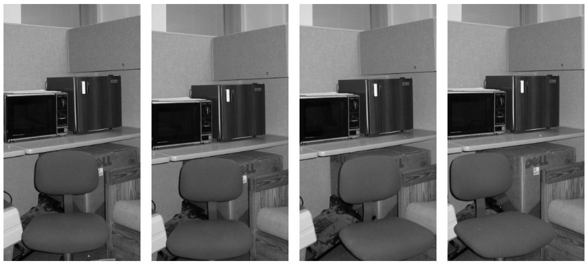

Fig. 4. Some of the training images used to generate the background eigenspace.

therefore the angle between the two spaces must not be small. However, even if the foreground and background image spaces are nearly orthogonal, it is possible (and even likely) that the angle between the subspaces of corresponding image sub-regions be small: in other words, the two spaces can be locally correlated even if globally they are not. Therefore, quasi-orthogonality of the two eigenspaces does not render the problem trivial. The computation of the mask is a crucial element of the algorithm. In principle, given estimates of the foreground and background components, the mask can be derived by direct minimization of the image reconstruction error. However, the discrete nature and large number of mask variables make direct minimization impractical—thus the need for the heuristic approach based on graph-cuts. In the definition of the image graph we used the simplest possible arc weight measure: difference of pixel intensity. More sophisticated measures are conceivable–for example, derived from correlations between image windows. Another possibility is to compute the gradient of the input image and encourage cuts at the locations where the gradient magnitude is large. Since our algorithm finds an approximate solution to the nonlinear system of equations (2) by an iterative process, the question of its convergence arises. In our experiments, the algorithm has always converged. We found that while the initial mask estimate is very rough and noisy, it improves gradually, and after 10-15 iteration steps, stops changing significantly. Plots of the decreasing reconstruction error (3) during the iteration process are shown on the right in Figs. 5 and 6. The proposed scheme inherits the fundamental limitation of the

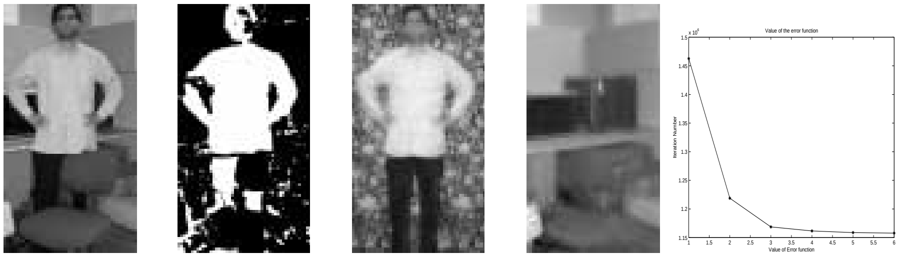

Fig. 5. Segmentation and de-occlusion at the end of six iterations. Left to right: 1. Input Image 2. Mask indicating foreground regions 3. Foreground reconstruction 4. Background reconstruction 5. Error (Equation 3) vs. iteration index

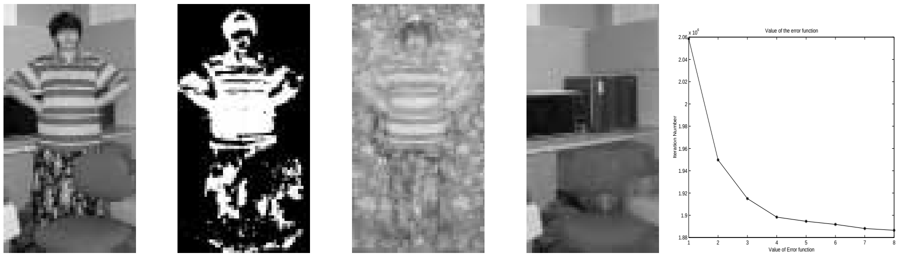

Fig. 6. Segmentation and de-occlusion at the end of eight iterations. Left to right: 1. Input Image 2. Mask indicating foreground regions 3. Foreground reconstruction 4. Background reconstruction 5. Error (Equation 3) vs. iteration index

eigenspace-based image representation: the location and pose of the objects in the test image can have only small variations with respect to the training set. However, there has been work [12] on achieving a degree of invariance with the eigenspace approach, and these schemes can be incorporated in our method. Another limitation of the method is that for input images where the spatial extents the the two components (foreground and background) are very different, the much larger component will dominate and the iterations may not converge on the correct solution. The proposed method is readily generalizable to $N > 2$ eigenspaces, thus modeling N groups of objects and their mutual occlusions. The difficulty that will have to be addressed is the computation of the mask, which will be N-valued, not binary. An immediate generalization is the inclusion of color information, which should result in much-improved masks. One could separately compute masks in the red, green, and blue channels and then merge them using, say, a majority vote. An approach that would not require color information would be the extraction of several 'channels'—for example, responses of Gabor filters at various scales and orientations—from the intensity images. At each iteration, the algorithm would derive a mask in each channel; the masks obtained from different channels can be combined (using a majority vote) and the resulting mask could be used in the next iteration. A more fundamental gen-

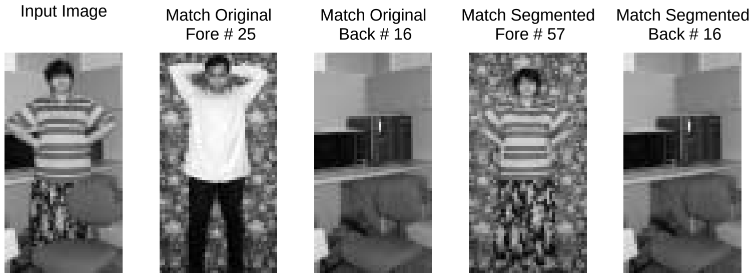

Fig. 7. Nearest neighbor matches: Left to right: 1. Input Image. 2. Nearest neighbor from foreground image set, using the original input image 3. Nearest neighbor from the background image set, using the original input image 4. Nearest neighbor from the foreground image set, using the foreground component extracted from the input image 5. Nearest neighbor from the background image set, using the background component extracted from the input image.

eralization is to spaces other than eigenspaces. One possibility is the exploration of the bases obtained by non-negative matrix factorization [13].

Relation to other approaches The work relating to occlusion removal reported in the series of papers by Leonardis and Bischof [5, 7, 6] is the most closely related to our method. We however, solve a different problem: we do not simply remove a nuisance occluder, but we use one eigenspace model for each group of objects of interest, and recover and reconstruct them from the input image. In addition, we introduce a separation mask that assigns pixels to the two components, and by doing so we regularize the problem by imposing smoothness constraints on the mask. Another difference is that the algorithm we use to remove occlusions is different: rather than perform a search in the image, we iteratively minimize a nonlinear error function, and optimally assign pixels to components by finding a minimal cut in a graph.

Our algorithm has an interesting relationship to image segmentation. By generating a binary separation mask, the algorithm effectively segments a group of objects from the input image. The proposed method achieves segmentation not by a traditional bottom-up process of organizing smaller homogeneous segments into meaningful real world objects, but by the use of models for the objects of interest. In fact, the images used in our experiments would be hard to segment by traditional methods. For example, the skirt in Fig. 3 would be very difficult to segment even manually. However, our algorithm generates a good segmentation because, in addition to information from the input image, it uses prior knowledge of the class of shapes it segments.

6 Summary and Conclusions

We introduced a method for segmenting an input image into foreground and background components, removing their mutual occlusions, and reconstructing

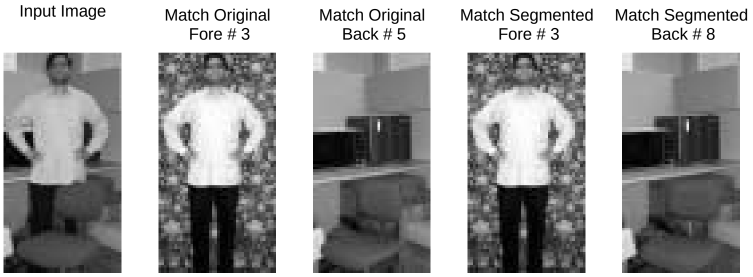

Fig. 8. Nearest neighbor matches: LEFT TO RIGHT: 1. Input Image. 2. Nearest neighbor from foreground image set, using the original input image 3. Nearest neighbor from the background image set, using the original input image 4. Nearest neighbor from the foreground image set, using the foreground component extracted from the input image 5. Nearest neighbor from the background image set, using the background component extracted from the input image.

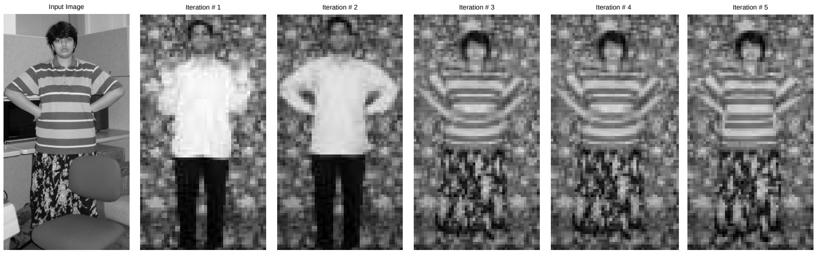

Fig. 9. Improvement in nearest neighbor matches over iterations: LEFT TO RIGHT: 1. The input image The nearest neighbor match obtained with: 2. The segmented image after the first iteration 3. The segmented image after the second iteration 4. The segmented image after the third iteration. 5. The segmented image after the fourth iteration 6. The segmented image after the fifth iteration. From the fifth iteration onwards the nearest neighbor match does not change for this example.

them. The segmentation is not obtained in the usual bottom-up manner, but is model-guided. The problem is formulated as a nonlinear system of equations, and the solution is approximated by an iterative process that alternates between segmentation and reconstruction, gradually improving the two components extracted from the input image. Once segmented from the input image and reconstructed, the image of the object of interest can be passed on to an object recognition module, substantially increasing recognition performance compared to the original input image. The method was successfully tested on segmenting, de-occluding, reconstructing, and recognizing humans in an office environment.

References

1. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 888–905
2. Malik, J., Belongie, S., Leung, T.K., Shi, J.: Contour and texture analysis for image segmentation. International Journal of Computer Vision **43** (2001) 7–27
3. Turk, M., Pentland, A.: Face recognition using eigenfaces. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Maui, Hawaii (1991)
4. Murase, H., Nayar, S.K.: Visual learning and recognition of 3-D objects from appearance. International Journal of Computer Vision **14** (1995) 5–24
5. Leonardis, A., Bischof, H.: Dealing with occlusions in the eigenspace approach. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA (1996) 453–458
6. Leonardis, A., Bischof, H.: Robust recognition using eigenimages. Computer Vision and Image Understanding: CVIU **78** (2000) 99–118
7. Bischof, H., Leonardis, A.: Robust recognition of scaled eigenimages through a hierarchical approach. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (1998) 664–670
8. Hadjidemetriou, E., Nayar, S.K.: Appearance matching with partial data. In: Proceedings of DARPA Image Understanding Workshop, Monterey, CA (1998)
9. Paulus, D., Drexler, C., Reinhold, M., Zobel, M., Denzler, J.: Active computer vision system. In Cantoni, V., Guerra, C., eds.: Computer Architectures for Machine Perception, Los Alamitos, California, USA, IEEE Computer Society (2000) 18–27
10. Huttenlocher, D., Lilien, R., Olson, C.: View-based recognition using an eigenspace approximation to the Hausdorff measure. IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (1999) 951–955
11. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. In: Energy Minimization Methods in Computer Vision and Pattern Recognition. (2001) 359–374
12. Black, M.J., Jepson, A.D.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. International Journal of Computer Vision (1998)
13. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401** (1999) 788–791