Translation, Scale and Occlusion-Tolerant Recognition with Multiple Eigenspace Models

Arnab Dhua, Florin Cutzu, Durgesh Dewoolkar
Conputer Science Department
Indiana University
Bloomington, IN 47405, USA

Stephen Kiselewich
Advanced Engineering Department
Delphi Corporation
Kokomo, IN 46904, USA

Abstract

We present a method for estimating the position and scale of occluder objects present in images of any of a number of modeled scenes. The background scenes are modeled using eigenspaces. If models of possible occluder objects are available we can then classify the detected foreground objects based on the available models. A segmentation of the object from the scene is also obtained in the process. We further handle the case when the foreground object region thus detected actually consists of mutually occluding objects and we locate, segment, de-occlude and recognize each object individually. We present numerical experiments to prove the validity of the method and demonstrate the utility of the algorithm in the detection, de-occlusion, segmentation and recognition of multiple objects in an office environment.

1. Introduction and background

We address the problem of recognizing objects included in a known scene regardless of location, scale, or occlusions. More precisely, we are given: (i) an eigenspace model for the background scenes (ii) a set of object eigenspaces, one for each of several foreground objects that can be placed at various locations within a given scene. The objects may occlude one another. Given an image of a scene containing several foreground objects, our goals are: (i) reconstruct the background scene by removing the occlusions created by the foreground objects, and (ii) detect and recognize the foreground objects and reconstruct their missing parts caused by occlusions.

The applicability of the eigenspace model [14] to 3-D object recognition has been limited by its limited intrinsic ability of coping with occlusion or image transforms like translation, rotation and scaling. The viewpoint variability problem can be solved by using enough views. Murase and Nayar improved on this approach by adding structure to eigenspace, joining the viewpoints according to their neighborhood relations on the viewing sphere, thus obtaining a discrete approximation of the views manifold of the object [10]. Distance to the manifold is a better recognition criterion than distance to eigenspace. The algorithm presented in this paper could easily be modified to incorporate this measure. Pentland, Moghaddam and Starner approach the problem of face recognition under general viewing conditions with a view-based multiple-observer eigenspace technique [12]. Given images of faces of N individuals under M different views their approach was to build a view-based set of M separate eigenspaces, each capturing the variation of the N individuals in a common view. The view-based eigenspace is essentially an extension of the eigenface technique to multiple sets of eigenvectors, one for each combination of scale and orientation. The first-step is to determine the location and orientation of the target object by selecting the eigenspace which best describes the input image, by calculating the error (distance from face space metric) using each eigenspaces' eigenvectors. After this step, the image is encoded using the eigenvectors of that view-space and then recognized. Black and Jepson address general affine transformations [2]. They define a subspace constancy assumption for eigenspaces similar to the brightness constancy assumption. For eigenspaces it can be assumed that there is a view of the object, as represented by some linear combination of the basis vectors (represented coefficients c) and some parametric spatial distortion (denoted as a), such that pixels in the reconstruction have the same brightness as the corresponding pixels in the image. The recognition goal is then to find c and a which minimize this objective function, thus resulting in a continuous optimization problem as opposed to an exhaustive search. In addition, the authors proposed a multi-scale eigenspace representation and a coarse-to-fine matching strategy in order to account for large affine transformations between eigenspace and the image.

A method for dealing with occlusions in eigenspace-based object recognition was described in [8, 9]. 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. If the resulting reconstructed image is close enough to the input image, 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.

Bischof and Leonardis [1] showed that their method can be applied to convolved and sub-sampled images yielding the same value of coefficients. Using the robust method proposed by the authors the coefficients need to be calculated at only one resolution (generally the lowest resolution) and can just be refined at higher resolutions. The authors also estimate the scale along with the coefficients. Starting with an initial estimate for coefficients, they minimize, with respect to the scale factor, the squared difference between the input image scaled by a factor and the reconstruction from the computed coefficients. Using the scale estimate the coefficient vector is re-estimated. This process continues till the convergence is reached. This step is repeated for each level in the eigenspace pyramid.

In [5] Hadjidemetriou and Nayar propose improvements to [8]. 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 [11]. They propose that the random selection scheme in [8] 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 [6]. 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. Previous work

In the present paper we build on previous work. In [3] we addressed a related problem: given an eigenspace model for the background scenes, an eigenspace model for the foreground objects, and an input image of a background scene containing a foreground object at a known location and scale but partially occluded by the background, we showed how to reconstruct the occluded portions of the background and object, and recognize both background and the foreground object. We briefly review the aspects of that work that are directly relevant to the present paper. Let $\boldsymbol{F}, \boldsymbol{B}$ be the background and foreground eigenspaces. The input image $\boldsymbol{g}$ is produced by two mutually occluding components, foreground $\boldsymbol{f} \in \boldsymbol{F}$ and background $\boldsymbol{b} \in \boldsymbol{B}$. The two components are combined using an unknown 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 i, $\boldsymbol{m}(i) = 1$ then only the foreground component is visible; if $\boldsymbol{m}(i) = 0$ only the background component is visible. Thus: $\boldsymbol{g} = \boldsymbol{M}\boldsymbol{f} + (\boldsymbol{I} - \boldsymbol{M})\boldsymbol{b}$, where $\boldsymbol{M}$ is a diagonal matrix with the vector $\boldsymbol{m}$ on the diagonal and $\boldsymbol{I}$ is the unit matrix. If ϕ, β are the unknown coordinate vectors of $\boldsymbol{f}$ and $\boldsymbol{b}$ in $\boldsymbol{F}$, $\boldsymbol{B}$, and the (known) means in the two spaces, respectively, $\bar{\boldsymbol{f}}$ and $\bar{\boldsymbol{b}}$, then the relation above becomes:

$$\boldsymbol{g} = \boldsymbol{M}(\boldsymbol{F}\phi + \bar{\boldsymbol{f}}) + (\boldsymbol{I} - \boldsymbol{M})(\boldsymbol{B}\beta + \bar{\boldsymbol{b}}). \qquad (1)$$

Our problem reduced to estimating ϕ, β and $\boldsymbol{M}$ given $\boldsymbol{g}$, $\boldsymbol{F}$, $\boldsymbol{B}$, and $\bar{\boldsymbol{f}}$ and $\bar{\boldsymbol{b}}$. This problem was under-determined, and therefore we imposed smoothness constraints on the mask. The steps of the algorithm, call it **Algorithm A**, were:

(1) Obtain initial estimates of the foreground: $\hat{\boldsymbol{f}} = \boldsymbol{F}\boldsymbol{F}^T(\boldsymbol{x} - \bar{\boldsymbol{f}}) + \bar{\boldsymbol{f}}$, and background $\hat{\boldsymbol{b}} = \boldsymbol{B}\boldsymbol{B}^T(\boldsymbol{x} - \bar{\boldsymbol{b}}) + \bar{\boldsymbol{b}}$.

(2) Estimate the mask $\hat{\boldsymbol{m}}$ by comparing locally the original image $\boldsymbol{x}$ to $\hat{\boldsymbol{f}}, \hat{\boldsymbol{b}}$. Generally, subject to smoothness constraints, if at pixel i the $\boldsymbol{x}$ 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 actual computation of $\hat{\boldsymbol{m}}$ was carried out by finding a minimum cut in the image graph, where the sink and source nodes correspond to the estimated foreground and background.

(3) Using the estimated mask $\hat{\boldsymbol{M}}$, re-estimate the foreground and background components: $(\hat{\phi}, \hat{\beta}) = \arg\min_{\phi,\beta} \|\boldsymbol{g} - \hat{\boldsymbol{M}}(\boldsymbol{F}\phi + \bar{\boldsymbol{f}}) - (\boldsymbol{I} - \hat{\boldsymbol{M}})(\boldsymbol{B}\beta + \bar{\boldsymbol{b}})\|_1$. Then, $\hat{\boldsymbol{f}} = \boldsymbol{F}\hat{\phi} + \bar{\boldsymbol{f}}$ and $\hat{\boldsymbol{b}} = \boldsymbol{B}\hat{\beta} + \bar{\boldsymbol{b}}$.

(3) Go to Step 2, or stop when changes in $\hat{f}, \hat{b}$ are small.

Thus, algorithm **A**, given the input image, extracts and de-occludes its foreground and background components.

3. Method

Our method of dealing with translation and scale variation requires the construction of a single background eigenspace, $\boldsymbol{B}$, using sample images of the different background scenes in which the foreground objects can be encountered. Let the n possible foreground objects be denoted by $O_1, \ldots, O_n$. The individual eigenspaces of the foreground objects, $F_1, \ldots, F_n$ are obtained by taking different views of each object. In addition to the eigenvectors, the corresponding mean images are also retained.

Given an input image, we first detect the region of the input image that corresponds to the occluding foreground ob-

ject and thus cannot be explained by $\boldsymbol{B}$ (Section3.1). Then, various object models F_k are tested against this region and the best-fitting models are determined, segmented, reconstructed, and recognized (Section 3.2). We now describe the method in detail.

3.1 Detecting the occluding objects

The goal at this stage is the detection of the image region corresponding to the occluder, "foreground", object(s). Since the identity, scale, and location of the foreign object is not known, the algorithm described in Section 2 cannot be applied without modifications: we do not have a foreground space $\boldsymbol{F}$; we only have the background eigenspace $\boldsymbol{B}$.

Below, the input image is $\boldsymbol{g}$; the occlusion mask $\boldsymbol{m}(i) = 1$ if pixel $i \in$ background and $\boldsymbol{m}(i) = 0$ if $i \in$ occluder; $\boldsymbol{M}$ is a diagonal matrix with $\boldsymbol{m}$ on the diagonal; $\boldsymbol{B}$ is the background eigenspace; β are the eigenspace coordinates of the background part of the input image; $\bar{\boldsymbol{b}}$ is the mean background image. Then:

$$\boldsymbol{M}\boldsymbol{g} = \boldsymbol{M}(\boldsymbol{B}\boldsymbol{\beta} + \bar{\boldsymbol{b}}). \quad (2)$$

We must solve for $\boldsymbol{M}$ and β. This is achieved by way of the following algorithm, call it **Algorithm B**. First, we project input image $\boldsymbol{g}$ on the background eigenspace, obtaining an initial estimate of the background component: $\hat{\boldsymbol{b}} = \boldsymbol{B}\boldsymbol{B}^T(\boldsymbol{g} - \bar{\boldsymbol{b}}) + \bar{\boldsymbol{b}}$. Then iterate the following two steps, until convergence:

(1) Use $\hat{\boldsymbol{b}}$ to estimate the mask: if at pixel i, $|\boldsymbol{g}(i) - \bar{\boldsymbol{b}}(i)| < \theta_t$, then $\hat{\boldsymbol{m}}(i) = 1$; else $\hat{\boldsymbol{m}}(i) = 0$. The threshold θ_t is determined automatically at every iteration t by using the fact that the histogram of the pixel reconstruction errors, $\boldsymbol{P}(|\boldsymbol{g}(i) - \bar{\boldsymbol{b}}(i)|)$, is bimodal–one mode, at low error levels, corresponds to background, and one mode, at high error levels, corresponds to the occluder [13, 4].

(2) Use the estimated mask to re-estimate $\hat{\boldsymbol{b}}$: $\hat{\beta} = \arg\min_\beta \|\hat{\boldsymbol{M}}\boldsymbol{g} - \hat{\boldsymbol{M}}(\boldsymbol{B}\beta + \bar{\boldsymbol{b}})\|$, and $\hat{\boldsymbol{b}} = \hat{\beta}\boldsymbol{B} + \bar{\boldsymbol{b}}$.

Convergence is quickly reached, typically within 5 iterations.

Due to viewpoint and illumination variability the background scene in the input image does not exactly fit the background eigenspace. Therefore, the mask derived by the above algorithm is, typically, noisy: although it finds a large group of pixels (not necessarily connected) that roughly correspond to the foreground object, it incorrectly labels a number of pixels, scattered across the image. Therefore, before further use, the mask is cleaned. Since the foreground object is known to be blob-like, the marginal (row and column) densities of the mask pixels that correspond to the foreground object are unimodal and strongly peaked at the center of mass of the occluder. By thresholding (using the same histogram-based method, [13, 4]) the row and column histograms, a bounding box around the occluder is found. Let the bounding box be denoted by $\boldsymbol{D}$. Only the portion of the mask within the bounding box is kept. Let this sub-mask be denoted by $\boldsymbol{R}$.

3.2 Identifying the occluders

The bounding box $\boldsymbol{D}$ can contain one or more of the objects O_k, at unknown scales, partially occluding each other. Our goal is to identify all the objects contained in the $\boldsymbol{D}$, and remove their mutual occlusions.

We fit all models to $\boldsymbol{D}$. Due to illumination and viewpoint variability, it is possible that certain pixels of $\boldsymbol{R}$ in fact belong to the background. Therefore, it is possible that a certain model fits only a sub-region of $\boldsymbol{R}$. It is also possible that pixels outside $\boldsymbol{D}$ belong in fact to the occluder. Therefore, $\boldsymbol{D}$ is only used as a guideline in narrowing down the image search.

For clarity of exposition, we present the case where only one object is present within $\boldsymbol{D}$. Consider model O_k. Its eigenspace is F_k. To fit it to the image, the eigenspace must be re-scaled. $\boldsymbol{D}$ is used a guideline in setting a range of possible scales and locations for O_k.

First, we scale F_k so that the training-set average width of O_k is equal to the width of $\boldsymbol{D}$. If height$(F_k) <$ height$(\boldsymbol{D})$, we try all possible locations of (the rescaled) F_k within $\boldsymbol{D}$. At each location, we apply algorithm **A**, described in Section 2, and then we determine the quality of the match as described below. If height$(F_k) >$ height$(\boldsymbol{D})$, we center the rescaled F_k on $\boldsymbol{D}$ and apply algorithm **A**.

The process is then repeated by using the height of $\boldsymbol{D}$ as the guideline i.e., we rescale F_k so that the training-set average height of O_k is equal to the height of $\boldsymbol{D}$. As before we try different placements of the scaled F_k within $\boldsymbol{D}$.

For each position of the rescaled F_k obtained using the height and width as guidelines, we derive new occlusion masks and reconstructions of O_k. The goodness score for the (position, scale) pair is defined such that it increases with the quality of the reconstruction of the input image using F_k, and decreases with the number of pixels in $\boldsymbol{R}$ left un-explained by F_k. Using this measure we find the best position and scale combination in the set of potential (position, scale) pairs.

Next, starting from the values for scale and position that maximize the goodness score, the fit of F_k to the image is further increased. This is done by using an iterative process in which we alternate between improving the location and scale fit by locally searching in scale and location ($\pm 10\%$ steps). These iterations converge to a local maximum of the goodness score.

Once the best position and scale of each model O_k are determined, the goodness scores of the various models are compared, to determine the object that best explains the input image.

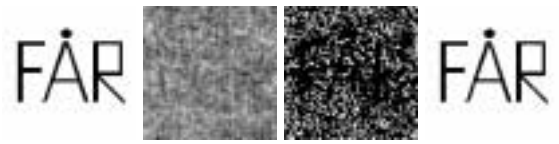

Figure 1: Testing algorithm **B**. From left to right: 1. Mask. 2. Input image 3. Recovered mask, iteration #1. 4. Recovered mask, iteration #4.

In general, more than one object is present in the bounding box $\boldsymbol{D}$. The procedure for multiple inter-occluding objects is, initially, as described above. At the end of the above process, we delete from the mask $\boldsymbol{R}$ the pixels that were explained by the recognized model. If enough pixels remain in $\boldsymbol{R}$ (say, $\geq 2\%$ of all image pixels) we continue the process and detect another object that gives the best fit to the remaining pixels in the mask $\boldsymbol{R}$.

At the conclusion of the process, the occluded parts of the foreground objects are reconstructed and identified, and the parts of the background hidden by the foreground objects are restored.

4. Experiments

4.1 Artificial images

We tested algorithm **B** on artificial images. We used the MIT Vision Texture database for generating foreground and background texture eigenspaces. The background eigenspace, $\boldsymbol{B}$ was derived from artificial textures, and the foreground eigenspace $\boldsymbol{F}$ from natural textures. All images were 128×128 pixels. Each eigenspace had 100 components; the angle between the two eigenspaces was 86°, indicating global, but not necessarily local, lack of correlation between the texture images. A background image b was generated using the basis $\boldsymbol{B}$, and a foreground image, f, was generated using the basis $\boldsymbol{F}$. The input image x was generated combining b and f using the binary occlusion mask shown on the left in Fig. 1: the black pixels in the mask were replaced with the corresponding pixels of the foreground image, and the white mask pixels were replaced with the corresponding pixels of the background image. Algorithm **B** was given as inputs the background eigenspace $\boldsymbol{B}$ and the input image. Even though the foreground eigenspace $\boldsymbol{F}$ was unknown, the algorithm quickly recovered the correct mask, as can be seen in Figure 1.

4.2 Real-world images

Our experimental setup consisted of two office scenes as backgrounds and three soft toys (a giraffe, a leopard and a lion) used as foreground objects. The toys were placed in the office background scenes to form the test images. The goal was to detect, segment, de-occlude the foreground objects in the scene and recognize them as one of the three toys.

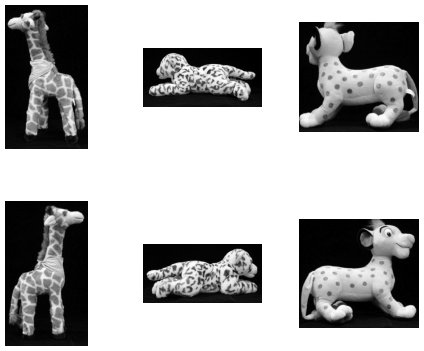

Figure 2: Examples of training images used for creating the object spaces: Each column shows the two extreme positions of each animal. COLUMN 1: Giraffe. COLUMN 2: Leopard. COLUMN 3: Lion.

We took 25 training images of each scene to construct the background eigenspace. The training images were taken with the camera panning by approximately $\pm 10^\circ$. Each background scene was 480×640 pixels in size.

We took 27 training images of each of the three soft toys. The training images covered the subject rotating $\pm 90^\circ$ about the mean frontal position, thus accounting for 180° of rotation about the vertical axis of the subject. As the objects were not completely rigid the positions of the limbs varied during acquisition. We did not restrict the possible movements of the limbs. The images of the soft toys were taken at a resolution of 480×640. We determined the tightest bounding box encompassing all views of a given foreground object. The size of the eigenspace for each object was determined by the size of this bounding box. The sizes of the images used for creating the eigenspace for each foreground object were 205×361 for the giraffe, 399×195 for the leopard and 269×247 for the lion.

The images were taken without attempting to make the lighting conditions identical in both background scenes. We did however, try to minimize the number of shadows cast on the scene.

We retained 90% of the variance in the background eigenspace which gave us only 4 eigenvectors for the background eigenspace, for the 50 training images used. The small number of eigenvectors was caused by the fact that the variation between the two scenes is much larger than the variation within each scene and this causes the largest component to be very dominant.

Similarly, we retained 90% of the variance for the object spaces, which gave us 10 for the giraffe, 15 for the leopard and 11 eigenvectors for the lion.

The test images were taken with roughly the same cam-

era location as the background training images with the toys placed at different positions in the scene. In both scenes we took a few test images with just one object and a few other images with multiple objects occluding each other. As before, the images are 480×640 pixels in size.

4.2.1 Object detection results

Sample results of the foreground object detection algorithm are shown in Figure 3. In the images shown here the red pixels indicate the mask ($\boldsymbol{R}$ in Section 3.1) and the green box is the detected bounding box ($\boldsymbol{D}$ in Section 3.1) encompassing the object. As can be seen in the figures, for the most part the masks and bounding boxes are quite accurate. However, in some images, parts of the foreground object are outside the bounding box, for example, in the second image in Figure 3 the nose of the giraffe has been excluded. Also, in some images, parts of the background are included in the mask region, for example, in the second image in Figure 3 the shadow of the giraffe was also included in the mask region.

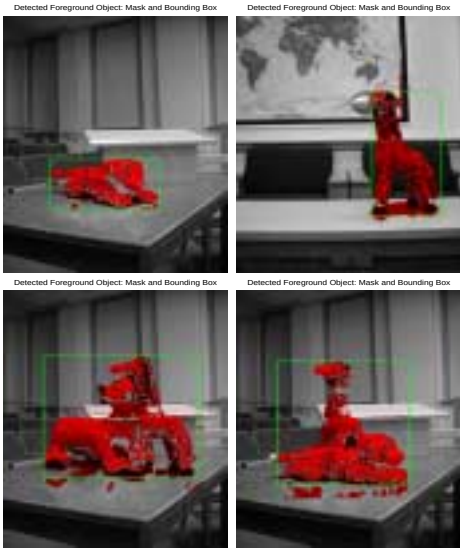

Figure 3: Detection of foreground object mask and bounding box, obtained by applying algorithm **B**. The red pixels indicate the foreign object mask $\boldsymbol{R}$ and the rectangles in green indicate the bounding boxes $\boldsymbol{D}$ (see Section 3.1).

4.2.2 Recognition results

Our goal was to detect a foreign object in the scene, determine its scale and location in the image, de-occlude it, segment it from the background, and finally recognize it.

Examples of the results obtained by the recognition algorithm described in Section 3.2 are shown in Figures 4–9. Figures 4 and 5 show the detection of a single object in each of the two scenes. Figures 6 and 7 show the detection of the lion and the giraffe appearing together in the same scene. Figures 8 and 9 show the detection of the leopard and the giraffe appearing together in the same scene. As can be seen, the foreground reconstruction contains the correct object in approximately the correct pose.

In the case of test images with only a single foreign object in the scene the algorithm was correctly able to recognize and segment 11 out of 12 such test images. For the case of test images with two mutually occluding foreign objects in the scene the algorithm was able to correctly recognize and segment 12 of the 16 objects present in a total of eight images.

An example of the kind of errors we get during the reconstructions is shown in Figure 10. As we can see, the neck and head of the giraffe have also been included in the foreground object mask obtained for the lion. During the deletion of the lion from the mask these pixels are removed and at the next step there are insufficient pixels to classify another object. So the giraffe ends up remaining undetected.

In Figures 4–9 the background reconstructions are accurate and the foreground object segmentation masks correspond to the object locations.

5 Discussion

We presented a method for dealing with occlusions and viewing transformations (translation, rotation and scaling) of an object of interest, included in known scene, using eigenspace models, without using an exhaustive search.

Leonardis and Bischof, in their series of papers [8, 9, 1], address the same problem using a multi-resolution approach and searching exhaustively at the lowest resolution. Their method is based upon a hypothesize-and-test paradigm with random selection of the hypotheses. This random approach might not be reliable, as suggested by Hadjidemetriou and Nayar [5], who specify the criteria for selecting hypotheses that maximize recognition rates. Our approach is different in the sense that although it uses a subset of the input image to calculate the coefficients, the subset used is not randomly selected. A structured approach is used to determine an occlusion mask that defines the subset of the image that should be used to compute the coefficients. This mask is refined using an iterative algorithm. The introduction of the mask is not just a computational aid; the mask embodies prior knowledge about the unknown occluders–for example, connectivity, or blob-like shape. In our test images, the test objects were placed in a real world, 3-D scene. In addition, our test images include shadows cast by the test objects and have variations in illumination conditions. Our algorithm was able to cope with these real world problems with reasonable success.

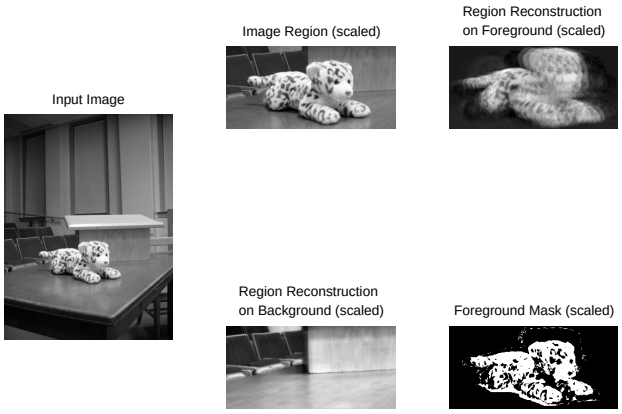

Figure 4: Recognition of foreground object and determination of segmentation mask: LEFT COLUMN: Input image from first scene. MIDDLE COLUMN, TOP: Image region where foreground object was detected. RIGHT COLUMN, TOP: Reconstruction on the detected object space. MIDDLE COLUMN, BOTTOM: Background behind the detected object. RIGHT COLUMN, BOTTOM: Segmentation of the recognized object.

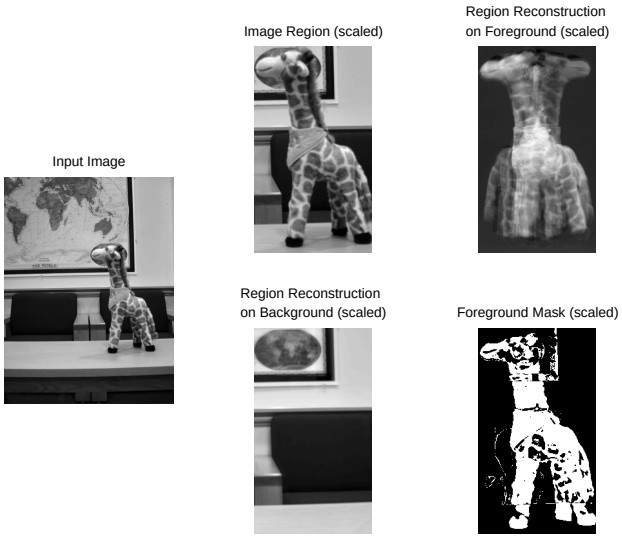

Figure 5: Recognition of foreground object and determination of segmentation mask: LEFT COLUMN: Input image from second scene. MIDDLE COLUMN, TOP: Image region where foreground object was detected. RIGHT COLUMN, TOP: Reconstruction on the detected object space. MIDDLE COLUMN, BOTTOM: Background behind the detected object. RIGHT COLUMN, BOTTOM: Segmentation of the recognized object.

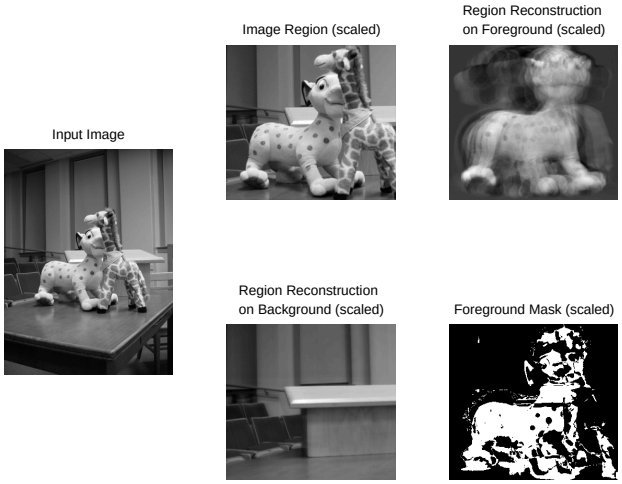

Figure 6: Recognition of foreground object and determination of segmentation mask: LEFT COLUMN: Input image containing multiple objects. MIDDLE COLUMN, TOP: Image region where first foreground object was detected. RIGHT COLUMN, TOP: Reconstruction on the detected object space. MIDDLE COLUMN, BOTTOM: Background behind the detected object. RIGHT COLUMN, BOTTOM: Segmentation of the recognized object.

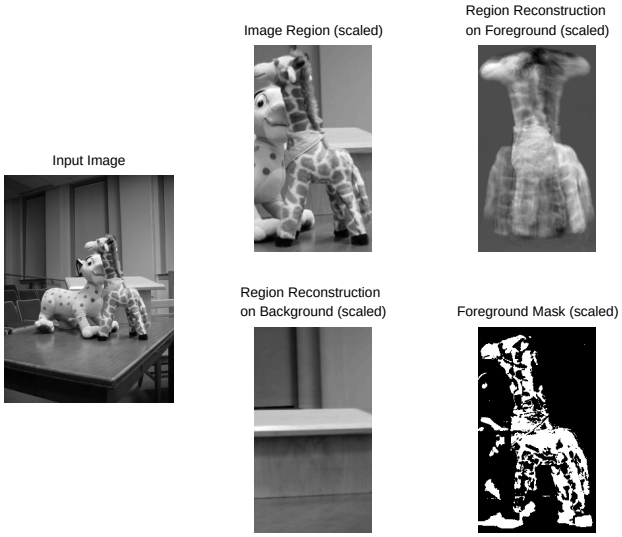

Figure 7: Recognition of foreground object and determination of segmentation mask: LEFT COLUMN: Input image containing multiple objects. MIDDLE COLUMN, TOP: Image region where second foreground object was detected. RIGHT COLUMN, TOP: Reconstruction on the detected object space. MIDDLE COLUMN, BOTTOM: Background behind the detected object. RIGHT COLUMN, BOTTOM: Segmentation of the recognized object.

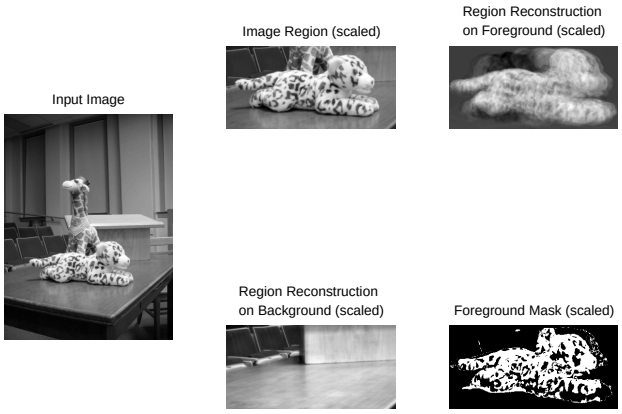

Figure 8: Recognition of foreground object and determination of segmentation mask: LEFT COLUMN: Input image containing multiple objects. MIDDLE COLUMN, TOP: Image region where first foreground object was detected. RIGHT COLUMN, TOP: Reconstruction on the detected object space. MIDDLE COLUMN, BOTTOM: Background behind the detected object. RIGHT COLUMN, BOTTOM: Segmentation of the recognized object.

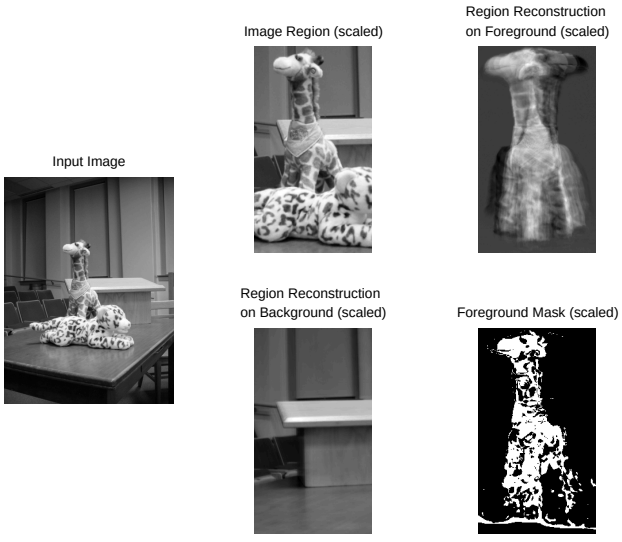

Figure 9: Recognition of foreground object and determination of segmentation mask: LEFT COLUMN: Input image containing multiple objects. MIDDLE COLUMN, TOP: Image region where second foreground object was detected. RIGHT COLUMN, TOP: Reconstruction on the detected object space. MIDDLE COLUMN, BOTTOM: Background behind the detected object. RIGHT COLUMN, BOTTOM: Segmentation of the recognized object.

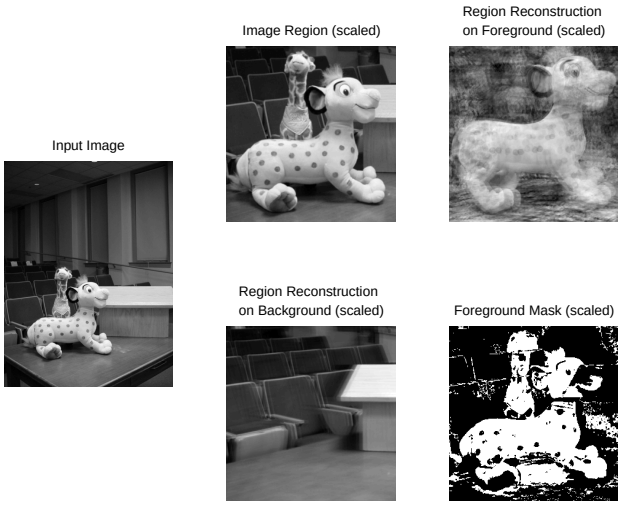

Figure 10: Example of error in recognition of foreground object and determination of segmentation mask: LEFT COLUMN: Input image containing multiple objects. MIDDLE COLUMN, TOP: Image region where foreground object was detected. RIGHT COLUMN, TOP: Reconstruction on the detected object space. MIDDLE COLUMN, BOTTOM: Background behind the detected object. RIGHT COLUMN, BOTTOM: Segmentation of the recognized object.

In their work on dealing with affine transformations in eigenspace models Black and Jepson [2] propose a robust error norm ρ to address the problem of outliers in the eigenspace representation and they define an outlier mask m based on a thresholding of the residual errors. We define a mask using a piecewise constancy assumption and we solve for the mask iteratively as shown in Section 2. The derived mask is useful for more than just outlier removal, it is actually the segmentation mask that can be used to separate the foreground regions from the background.

In the first step of our method (algorithm **B**) we use the pixel-wise differences of the input image to the reconstruction (in the background space) to arrive at an initial estimate of the location of the foreground object. This differs from the traditional eigenspace methods that use only the object eigenspace to locate the position of the foreground object. However, unlike other methods we also need to model the range of expected backgrounds.

Our algorithm uses an iterative framework to obtain the segmentation of the object of interest from cluttered background scenes. The methods discussed above do not perform this type of real world segmentation. The segmentation step is not just a byproduct of the method, but it also enhances the quality of the reconstructions and thereby the accuracy of the recognition. Note that 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.

A limitation of our method is that it tolerates only a limited amount of occlusion of the foreground objects by the background. If, say, only the giraffe's neck was occluded, its reconstruction and recognition would still be possible, as its bounding box computed by algorithm **B** would still be correct. However, if only its neck and head were visible, its reconstruction and recognition would fail. Another limitation is that in its current form, the method uses pixels as features, hence its sensitivity to illumination change. Therefore, it is important to generalize the method to spaces other than eigenspaces, and features other than pixels–for example, edge maps, as suggested in [6]. One possibility is the exploration of the bases obtained by non-negative matrix factorization [7].

Algorithm **A** 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. Our proposed method can benefit (in the reconstruction of both background and foreground, and in error-calculation steps) from the use of view and illumination eigenspace manifolds introduced in [10]. Our current view sets were too small to adequately sample the view manifolds. An interesting possibility is the replacement of the hard occlusion mask with of soft, or semi-transparent mixing masks. Thus, pixel i of the image is a mixture of the corresponding pixels generated by various models. 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.

6 Summary and conclusions

We present a recognition scheme that locates the position and scale of a foreground object in an input image. Once the foreground object has been roughly located we accurately determine its exact position using an object model and a local search method. In this process we obtain a segmentation mask for the scene. Moreover, the proposed scheme is able to correctly segment and recognize multiple objects that occlude each other. We present the successful application of our method to real world office scenes in locating, segmenting, de-occluding and recognizing 3D objects.

References

[1] H. Bischof and A. Leonardis. Robust recognition of scaled eigenimages through a hierarchical approach. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 664–670, June 1998.

[2] M. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *Proceedings of European Conference on Computer Vision*, pages 329–342, 1996.

[3] A. Dhua, F. Cutzu, D. Dewoolkar, and S. Kiselewich. Multiple eigenspace models for scene segmentation and occlusion removal. Technical report, Indiana Uinversity, Computer Science Department, 2004.

[4] R. Gonzalez and R. Woods. *Digital Image Processing*, chapter 10, pages 598–600. Prentice-Hall, 2002.

[5] E. Hadjidemetriou and S. K. Nayar. Appearance matching with partial data. In *Proceedings of DARPA Image Understanding Workshop*, Monterey, CA, November 1998.

[6] D. Huttenlocher, R. Lilien, and C. Olson. View-based recognition using an eigenspace approximation to the Hausdorff measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):951–955, September 1999.

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

[8] A. Leonardis and H. Bischof. Dealing with occlusions in the eigenspace approach. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 453–458, San Francisco, CA, June 1996.

[9] A. Leonardis and H. Bischof. Robust recognition using eigenimages. *Computer Vision and Image Understanding: CVIU*, 78(1):99–118, 2000.

[10] H. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14:5–24, 1995.

[11] D. Paulus, C. Drexler, M. Reinhold, M. Zobel, and J. Denzler. Active computer vision system. In *Computer Architectures for Machine Perception*, pages 18–27, Los Alamitos, California, USA, 2000. IEEE Computer Society.

[12] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, WA, June 1994.

[13] T.W. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *SMC*, 8(8):629–632, August 1978.

[14] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Maui, Hawaii, 1991.