Robust Eigendecomposition Methods for Object Pose Detection

Edmond M. DuPont, Hyun Geun Yu, and Rodney G. Roberts
Department of Electrical and Computer Engineering
FAMU-FSU College of Engineering
Tallahassee, Fl 32310, USA
edupont@eng.fsu.edu

***Abstract*— This work explores image processing techniques that involve the application of eigenspace methods for pose detection. An eigenspace method for data compression used in the image processing field is commonly referred to as Principal Component Analysis (PCA). We present some recently introduced eigenspace concepts for detecting the pose angle of an occluded object located in an image containing background clutter. To detect the pose of a target object in the presence of background and occlusions we analyze two eigendecomposition methods. The quad-tree structure includes dividing the training images into quadrants and creating a subspace eigendecomposition for each level. A statistical robust approach is also applied that weights the background and occlusion pixels based on their influence on the reconstruction of the desired target object. We review both of these pose detection approaches and illustrate each application with an example.**

Keywords—principal component analysis, robust PCA, pose detection, background, occlusion

1. Introduction

Some vision techniques are based on the various shapes or models of objects located within an image. Our research focuses primarily on the appearance of objects within an image. Detection of objects within an image containing other objects presents some difficulty in trying to locate the target object as well as detecting its pose. For our research we assume that a set of training images are obtained within a controlled environment with the absence of background and occlusions. Principal Component Analysis (PCA) techniques will be used as the main part of the image analysis algorithm.

PCA subspace methods have been used in many other vision applications such as object recognition [3], face detection [4], and visual object tracking [1], in addition to others. In this work PCA will be applied to a set of highly correlated images to reduce them to a lower dimensional eigenspace. This eigenspace is the span of a set of eigenimages whose linear combination can represent the appearance of all of the training images. This work gives an application of PCA techniques to the object pose detection. We will present a recently introduced eigenspace method for object localization. The quad-tree and robust M-estimation techniques are then applied based on candidate localization points of the target object within a test image. In terms of statistics, this quadtree method is based on the minimization of the least squares estimation. Due to the common robustness issues of the least squares method we also applied a robust M-estimation method. This robust pose detection method will effectively weight the pixels of the background and occlusions based on their influence to the solution. We implement these algorithms to observe their accuracy for pose detection in the presence of background and occluding objects.

2. Previous Work

Detecting objects along with their orientation within a test image of a real world scene is a difficult problem. This problem has been studied over the years to develop efficient algorithms for finding the correct pose of an object [3], [5], [6]. There are many different applications for which eigenspace methods for pose detection can be applied. Krumm [6] proposed a method that is based on using eigenspace methods to detect the appearance of features of an object as it is rotated. Wang and Ben-Arie [7] proposed a method to detect objects based on models of standard shapes. Their study focused on common man-made objects that appear within an environment. Eigenspace methods were used to detect the various appearances of model shapes, edges and boundaries that the objects may represent. This method becomes increasingly insufficient when the data set of all the possible model shapes becomes very large. One recent technique explored by Chang, et.al, [3] used PCA methods for object pose detection in a controlled environment. Their technique utilized an eigendecomposition of highly correlated images to reduce the computation of traditional eigenspace methods. Black and Jepson [1] applied statistical robust techniques for eigentracking to robustly compute outliers within a test image. These outliers relate to pixels within the test image that have minimum influence on the desired solution. Most of these concepts utilize eigenspace approximation methods to match the appearance of the object rather than shape to account for the possible variations within the object. Within an extremely controlled environment eigenspace algorithms tend to work well but in the presence of background clutter with or without occlusion, additional problems can occur. Some possible solutions to deal with these effects will be presented later. A common issue that arises in applying eigenspace methods is computational expense. As the image data set increases the calculation required to calculate all of the principal components also increases.

3. PRELIMINARIES

In this work all training and testing images used are grayscale images. Each image within the training data set consist of 256×256 pixels which are then "rowscanned" [3] into a vector **d**. The collection of training images are first placed into a matrix D with columns $\mathbf{d}_i$, for $i = 1, 2, \ldots, n$:

$$D = \left[\begin{array}{cccc} \mathbf{d}_1 & \mathbf{d}_2 & \cdots & \mathbf{d}_n \end{array}\right]. \tag{1}$$

The average image vector $\bar{\mathbf{d}}$ is defined as

$$\bar{\mathbf{d}} = \frac{1}{n}\sum_{j=1}^{n}\mathbf{d}_j \tag{2}$$

which is then placed in an average data matrix $\bar{D}$ denoted as

$$\bar{D} = \left[\begin{array}{cccc} \bar{\mathbf{d}} & \bar{\mathbf{d}} & \cdots & \bar{\mathbf{d}} \end{array}\right]. \tag{3}$$

This average data matrix $\bar{D}$ is created to subtract the average of all the training images from each of the training images $\mathbf{d}_i$ using the equation

$$\hat{D} = D - \bar{D}. \tag{4}$$

The objective is to be able to reconstruct the training images using a linear combination of the dominant eigenimages. Applying the Singular Value Decomposition to the data matrix $\hat{D}$ results in the equation

$$\hat{D} = U\Sigma V^T, \tag{5}$$

where U and V are orthogonal matrices. The matrix Σ is a diagonal-like matrix whose diagonal components are the square root of the variances [3]. The column vectors of the matrix U contain the eigenimages ordered in decreasing variance according to the diagonal components of the Σ matrix.

4. LOCALIZATION

For our test image the pixel size is assumed to be larger than the pixel size of the training images. For our test images we use test images of pixel size 512×512. We also assume the target object contains the same number of pixels as the training objects. In order to determine the pose of the object within the test image we need to be able to find possible locations of the target object. Chang, et.al., [3] proposed a method that calculated possible candidate locations of the target object within a test image through a sliding window technique. As this window slides it takes the current image and projects it onto the eigenspace spanned by the first *k* principal components. The measure of possible locations are evaluated by the following equations:

$$m_1 = \frac{m_{proj}}{\|\mathbf{y}\|}, \quad \text{with } m_{proj} = \sqrt{\sum_{i=1}^{k}(\mathbf{u}_i^T\mathbf{y})^2} \tag{6}$$

where **y** is the image vector corresponding to the sliding window and the vectors $\mathbf{u}_i$ correspond to the columns of the matrix U, which are the eigenimages. In the presence of occlusion and background clutter the calculation of measure m_1 identifies a number of candidate locations. To reduce this number a second measure m_2 was introduced. This second measure quantifies the change resulting from a vertical and horizontal shift of the window. Measure m_2 views the large changes as possible locations of the target object:

$$m_2 = \sqrt{\left(\frac{\partial^2 m_{proj}}{\partial v^2}\right)^2 + \left(\frac{\partial^2 m_{proj}}{\partial h^2}\right)^2}. \tag{7}$$

The measure m is a combination of the measures m_1 and m_2 to give a set of final candidate locations:

$$m = \begin{cases} m_2 & \text{if } m_1 \geq \rho \\ 0 & \text{if } m_1 < \rho \end{cases} \tag{8}$$

where ρ is a user-defined threshold value. This narrows the range of possible candidate locations. Once all the possible candidate locations have been calculated each location is then evaluated by being projected onto the eigenspace spanned by the *k* eigenimages.

5. QUAD-TREE POSE DETECTION

For the problem of dealing with background clutter and occlusion the online process becomes increasingly difficult. Due to the sensitivity of eigenspace methods, projecting the test image with the addition of background or occlusion onto the eigenspace can cause the projection point to fall far away from the desired point on the manifold. Intensity normalization of the test image with the effects of the background will alter the intensity values of the target object making it vary greatly from the intensity of the trained object within the training images. Scale normalization cannot be done due to the difficulty of finding the edges of the desired target object in the presence of occlusion or background clutter.

The candidate locations are then evaluated by applying a quad-tree structure of the training images to the test image window to find the pose of the object. The offline phase is divided into levels and at each level the training images are divided into quadrants. In general, for a given level *l* each training image is divided into $4^{(l-1)}$ sub-images. By creating these levels a larger image data matrix is created:

$$D_{l,j} = [\mathbf{d}_{1,l,j} \quad \mathbf{d}_{2,l,j} \quad \cdots \quad \mathbf{d}_{n,l,j}] \tag{9}$$

where *j* represents the sub-image in level *l*. For each level, a set of eigenimages and resulting manifolds are created from the set of training images.

For the online phase, the sliding windows within the test image are also divided into sub-images depending on the level. Initially at the first level the entire window is projected onto the eigenspace, and the normalized distance to the manifold is calculated using the following equation:

$$d = \frac{\|\mathbf{t}_\theta - \mathbf{p}\|}{\|\mathbf{t}_\theta\|}. \tag{10}$$

Two user defined thresholds are set at an 'Accept' and a 'Reject' value. If the measured distance d falls between the two threshold levels, then the algorithm advances to the next level. If within any of the sub-levels, a manifold distance is above the 'Reject' level, then that corresponding quadrant is not considered at the next sub-level. When the manifold distance reaches the 'Accept' level, the image number corresponding to the smallest distance is the detected pose. All sub-windows with an acceptance receives a positive vote. Detection of pose θ is achieved when the normalized distance of one or more sub-windows reaches a minimum user-defined threshold.

6. Robust Pose Detection

In this section we will consider the analysis of PCA techniques in a statistical least squares sense to incorporate robust techniques for object pose detection. In this case we consider the pixels corresponding to occlusion and background clutter as possible outliers that will affect the deviation of the projection of a test image onto a manifold of a training image data set. We present a weighted least squares method [1], [4] that applies a robust estimation technique that weights the pixels of a test image.

Recall that traditional PCA least squares approximation assumes that the test image is pre-processed similar to the training images and segmented from the background. However, the test image differs slightly from the training images due to some background or occlusion. Unfortunately, PCA least squares is sensitive to the errors due to background and occlusion which results in a significant error between the reconstructed image and the desired target image. To overcome this problem the standard quadratic error norm used in calculating the MSE is replaced with the robust energy function [4]:

$$\begin{aligned} E_{robust} &= \sum_{i=1}^{n} e_{robust}(\mathbf{d}_i - \boldsymbol{\mu} - B\mathbf{c}_i, \boldsymbol{\sigma}) \\ &= \sum_{i=1}^{n}\sum_{p=1}^{d} \rho(d_{pi} - \mu_p - \sum_{j=1}^{k} b_{pj}c_{ji}, \sigma_p) \end{aligned} \tag{11}$$

where σ_p is a parameter for each pixel, $\boldsymbol{\mu}$ is the average image vector, B is an orthogonal matrix containing the eigenvectors, and $C = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_n]$ is the matrix of coefficients that reconstruct the data. Additional details can be found in [4]. For the experiments shown here we used the Geman-McClure error function given by

$$\rho(x, \sigma) = \frac{x^2}{\sigma^2 + x^2} \tag{12}$$

for robust estimation [1], [2], [4]. Applying this robust estimator to the minimization cost function, we are able to weight the influence of each error pixel. This leads to the implementation of weighted least squares approximation to calculate a new set of coefficients based on the final error values. The solution to weighted least squares (WLS) is formulated by using the iteratively reweighted least-squares (IRLS) method. A weight matrix W is calculated in the IRLS method and is updated every iteration until the difference in the error values converges to a user defined threshold. Each element of the weight matrix W corresponds to an element of the error vector. As a result the elements of the weight matrix can be written as follows:

$$w_p = \frac{\psi(e_p, \sigma_p)}{e_p} \tag{13}$$

where $\psi(e_p, \sigma_p) = \frac{\partial\rho(x,\sigma)}{\partial x}$ is the influence function of the robust estimator, σ is a defined scale parameter, and e_p is the pixel value of the error vector. The scale parameter σ is used to measure the influence the outliers have on the data. The outlying process identifies the pixels with greater influence on the residuals by the following condition:

$$|e_p| > \frac{\sigma}{\sqrt{3}}. \tag{14}$$

Once convergence is achieved the resulting coefficients are projected onto the eigenspace spanned by the basis eigenimages for image reconstruction. This leads to the implementation of the robust weighted PCA of the IRLS method to account for background clutter and occlusions as outliers for pose detection.

7. Example

In this section we present an example to show how the online quad-tree approach evaluates each candidate location of the sliding window at each of the levels. This sliding window was projected onto the eigenspace spanned by the current level to detect the correct pose. The normalized distance 'Reject' and 'Accept' thresholds were set to 1.2 and 0.65, respectively. At level one, we evaluate the entire window as shown in Fig. 1, where the distance from at least one training image fell within the upper and lower thresholds. The 'Reject' and 'Accept' thresholds are denoted by the dotted and solid line, respectively. At level two the entire window was divided into quadrants. Each quadrant was then compared with its respective manifolds. The calculated normalized distances of the sub-windows at this level also produced at least one value between the thresholds. Each of the quadrants at level two were divided into sub-quadrants at level three depicted in Fig. 2. In level three, it was evident from the first three plots along the second row that the corresponding sub-images provided an adequate distance measure acceptable to their respective manifolds for a positive pose detection. All three sub-windows resulted in a correct pose detection related to the 48-th of 72 training images. This corresponds to pose $\theta = 240°$.

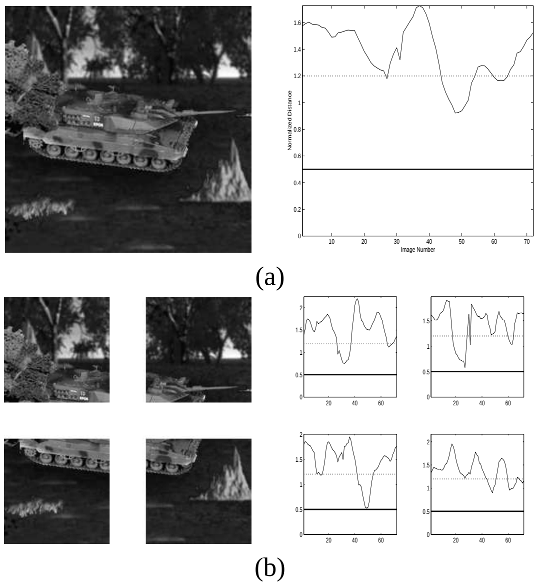

Figure 1. (a) Level one target window of first candidate location with distance plot from all the training images. (b) Level two quadrants with the corresponding plots of manifold distances. In both cases, at least one distance measured between the thresholds.

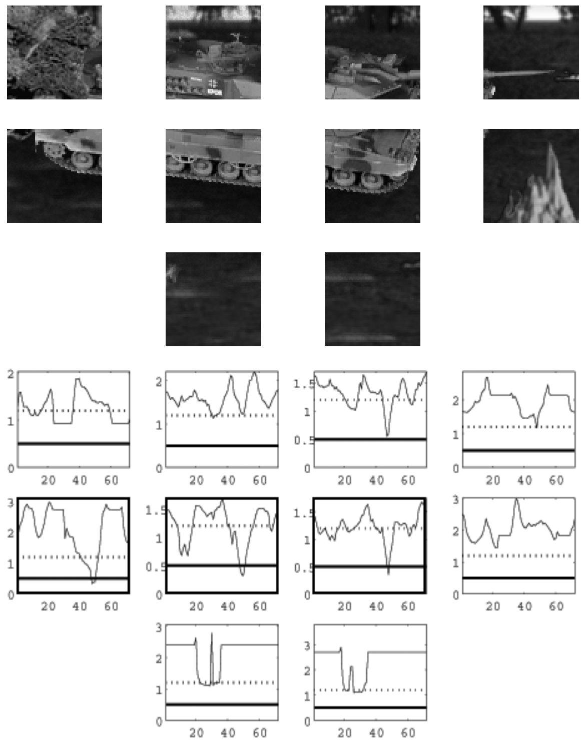

Figure 2. Expansion to level three produced these sub-images where the three highlighted plots were dominant in successful pose detection.

We also applied robust M-estimation techniques to the online process for pose detection. The image in Fig. 3a shows the sliding window containing the target object in addition to background and occlusion. The entire image was projected onto the subspace spanned by the eigenimages for reconstruction. The error image shown in Fig. 3b was calculated using the difference between the target image and its projection onto the eigenspace. We chose the scalar parameter $\sigma = 0.1294$ as the median of the absolute error vector to create the weight matrix W. Using IRLS we applied the robust estimator given in equation (11) and calculated the weight matrix W. Fig. 3c shows the resulting weight image that was calculated for the first iteration. As shown in the plot of the weight function in Fig. 3d, all of the background and occlusion pixels within the error images received a lower weight value than the pixels pertaining to the target object. As a result a new set of coefficients were calculated and used to find the error image for the next iteration. This iteration process is repeated until the difference between the normalized error converges. Once convergence was achieved the normalized distance of the final coefficients were measured against the manifold of the training images. Depending on the value chosen for σ, the effects of the background and occlusion will play a higher or lower role in the reconstruction. Adjusting the σ parameter to a higher value may tend to down-weigh some of the pixels of the target images. Conversely a σ value that is too low would not down-weigh enough background or occlusion for positive detection. With this robust method all pixels pertaining to the background and occlusion were down-weighted compared to the pixels of the target object.

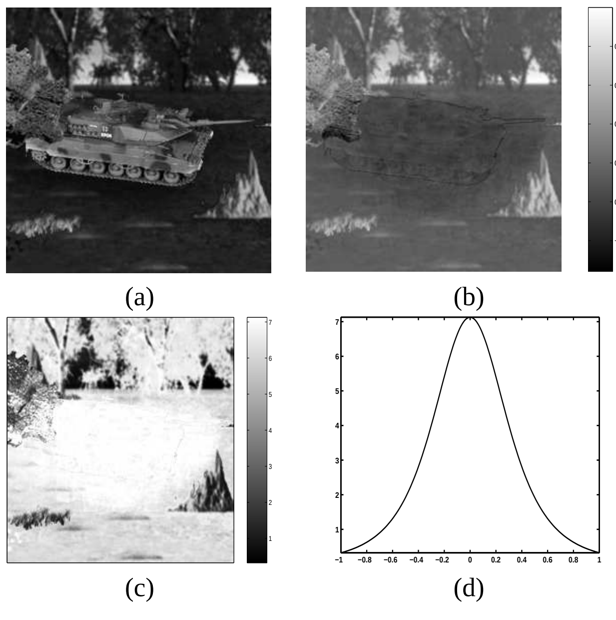

Figure 3. (a) The target image from the localization sliding window. (b) Error image **e**. (c) Weight image with lighter pixels receiving a higher weight that the darker pixels. (d) Plot of weight function used to weight each pixel of the error vector **e**.

8. Conclusion and Future Work

This paper examined two methods of applying eigendecomposition techniques for pose detection of non-planar objects within planar images in the presence of background clutter and occlusions. A set of highly correlated images of a 3-dimensional object as it was rotated about an axis was used as the training image data set. The proposed algorithms were implemented in Matlab to develop simulation results. With this implementation we can explore the advantages and disadvantages of the two approaches. Test image data sets are being developed that include images with variations of the intensity levels between the target image, background and occluding object. Computational timing results are also considered and compared between the two algorithms. The quadtree approach exhibits relatively fast online calculations as compared to the robust method. Experiments of the robust estimation method proved to take a significantly longer online computation time due to the fact that this method effectively evaluates each pixel within the test image window. We are currently conducting additional simulations relative to these problems and will report our results in subsequent submissions.

References

[1] M. Black and A. Jepson, "EigenTracking: Robust matching and tracking of articulated objects using a view-based representation," *Proc. Int. Conf. on Computer Vision*, 1998, pp. 63-84.

[2] M. Black and A. Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications in early vision," *Proc. Int. Conf. on Computer Vision*, 1996, pp. 57-91.

[3] C. Y. Chang, A. A. Maciejewski, V. Balakrishnan, and R. G. Roberts, "Eigendecomposition-based pose detection in the presence of occlusion," *IEEE/RSJ International Conference on Intelligent Robots and Systems, Volume: 1*, 29 Oct.-3 Nov. 2001, pp. 569-576.

[4] F. De La Torre and M. Black, "A framework for robust subspace learning," *Proc. Int. Conf. on Computer Vision*, 2001, pp. 1-8.

[5] M. Hiroshi and S. Nayar, "Visual learning and recognition of 3D objects from appearance," *International Journal of Computer Vision*, 1995, pp. 5-24.

[6] J. Krumm, "Eigenfeatures for planar pose measurement of partially occluded objects," *Proc. IEEE Computer Society Conference*,18-20 June 1996, pp. 55-60.

[7] Z. Wang and J. Ben-Arie, "Generic object detection using model based segmentation," *Proc. IEEE Computer Society Conference*, 23-25 June 1999, pp. 23-25.