Pattern Anal Applic (2007) 10:15–31
DOI 10.1007/s10044-006-0046-6

THEORETICAL ADVANCES

Quadtree-based eigendecomposition for pose estimation in the presence of occlusion and background clutter

Chu-Yin Chang · Anthony A. Maciejewski · Venkataramanan Balakrishnan · Rodney G. Roberts · Kishor Saitwal

Received: 25 February 2005 / Accepted: 6 June 2006 / Published online: 20 December 2006

Abstract Eigendecomposition-based techniques are popular for a number of computer vision problems, e.g., object and pose estimation, because they are purely appearance based and they require few on-line computations. Unfortunately, they also typically require an unobstructed view of the object whose pose is being detected. The presence of occlusion and background clutter precludes the use of the normalizations that are typically applied and significantly alters the appearance of the object under detection. This work presents an algorithm that is based on applying eigendecomposition to a quadtree representation of the image dataset used to describe the appearance of an object. This allows decisions concerning the pose of an object to be based on only those portions of the image in which the algorithm has determined that the object is not occluded. The accuracy and computational efficiency of the proposed approach is evaluated on 16 different objects with up to 50% of the object being occluded and on images of ships in a dockyard.

Keywords Singular value decomposition · Quadtree decomposition · Partial occlusion · Background clutter · Object recognition · Pose estimation

C.-Y. Chang
Energid Technologies, 124 Mount Auburn Street, Suite 200, North Cambridge, MA 02138, USA

A. A. Maciejewski · K. Saitwal (✉)
Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373, USA
e-mail: aam@colostate.edu

V. Balakrishnan
Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-1285, USA

R. G. Roberts
Department of Electrical and Computer Engineering, Florida A&M—Florida State University, Tallahassee, FL 32310-6046, USA

1 Introduction

One of the fundamental problems in computer vision is the recognition and localization of three-dimensional objects. Subspace methods represent one computationally efficient approach for dealing with this class of problems. Variously referred to as eigenspace methods, principal component analysis methods, and Karhunen–Loeve transformation methods [1, 2], these have been used extensively in a variety of applications such as face characterization [3, 4] and recognition [5–9], lip-reading [10, 11], object recognition [12–16], pose estimation [17–19], robot position estimation [19, 20], visual tracking [21, 22], and inspection [23–26]. All of these applications are based on taking advantage of the fact that a set of highly correlated images can be approximately represented by a small set of eigenimages [27]. Once the set of principal eigenimages is determined, online computation using these eigenimages can be performed very efficiently.

Unfortunately, one of the drawbacks associated with using eigendecomposition-based approaches is that they are very sensitive to occlusion and background clutter [24, 28–44]. The purpose of this work is to explore the feasibility of applying eigendecomposition to

a quadtree representation of correlated images in order to efficiently accommodate the presence of occlusion and background clutter. The pose estimation problem is used here as a representative application. In the next subsection, the fundamentals of applying eigendecomposition to related images are reviewed. This is followed by an overview of the standard approach to solving the pose estimation problem using eigendecomposition and a discussion of why occlusion and background clutter present such difficulty.

1.1 Eigendecomposition of related images

An image can be represented as an $h \times v$ array of square pixels with intensity values normalized between 0 and 1. Thus, an image will be represented by a matrix $\mathcal{X} \in [0,1]^{h \times v}$. Since we will be considering sets of related images, it will be convenient to represent an image equivalently as a vector, obtained simply by ''row-scanning'', i.e., concatenating the rows to obtain the *image vector* $\mathbf{x}$ of length $m = hv$:

$$\mathbf{x} = \mathrm{vec}(\mathcal{X}^{\mathrm{T}}).$$

The *image data matrix* of a set of images $\mathcal{X}_1, \ldots, \mathcal{X}_n$ is an $m \times n$ matrix, denoted X, and defined as

$$X = [\mathbf{x}_1 \ \cdots \ \mathbf{x}_n],$$

with typically $m \gg n$. We consider only the case where n is fixed, as opposed to cases where X is constantly updated with new images.

The *average image vector* is denoted $\bar{\mathbf{x}}$ and defined as

$$\bar{\mathbf{x}} = (\mathbf{x}_1 + \cdots + \mathbf{x}_n)/n.$$

The corresponding *average image data matrix*, denoted $\bar{X}$, is

$$\bar{X} = [\bar{\mathbf{x}} \ \cdots \ \bar{\mathbf{x}}].$$

The matrix $X - \bar{X}$, which we denote $\hat{X}$, has the interpretation of an ''unbiased'' image data matrix.

The singular value decomposition (SVD) of $\hat{X}$ is given by

$$\hat{X} = \hat{U}\hat{\Sigma}\hat{V}^{\mathrm{T}},$$

where $\hat{U} \in \mathbb{R}^{m \times m}$ and $\hat{V} \in \mathbb{R}^{n \times n}$ are orthogonal, and $\hat{\Sigma} \in \mathbb{R}^{m \times n}$, with $\hat{\Sigma} = [\hat{\Sigma}_d \ \mathbf{0}]^{\mathrm{T}}$, where $\hat{\Sigma}_d = \mathrm{diag}(\hat{\sigma}_1, \ldots, \hat{\sigma}_n)$, with $\hat{\sigma}_1 \geq \hat{\sigma}_2 \geq \cdots \geq \hat{\sigma}_n \geq 0$, and $\mathbf{0}$ is an n by $m - n$ zero matrix. The SVD of $\hat{X}$ plays a central role in several important imaging applications such as image compression, pattern recognition and pose estimation. The columns of $\hat{U}$, denoted $\hat{\mathbf{u}}_i$, $i = 1, \ldots, m$, are referred to as the eigenimages of $\hat{X}$; these can be interpreted as estimates of the eigenvectors of the covariance matrix of the image vector. The eigenimages provide an orthonormal basis for the columns of $\hat{X}$, ordered in terms of importance; the corresponding singular values measure how ''aligned'' the columns of $\hat{X}$ are with the associated eigenimage. The components of the ith column of $\hat{V}$ measure how much each individual image contributes to the ith eigenimage.

1.2 Eigendecomposition applied to pose estimation

The standard application of eigendecomposition to solve the pose estimation problem requires the computation of a reduced-order representation of the set of all possible orientations for the object being considered. For this purpose, several intermediate orientations of an object are normally used. Because the eigenspace representation of an image is very sensitive to changes in size and intensity, each training image is normalized to account for differences in scale and brightness. The average normalized training image is then subtracted from each of the normalized training images and the eigendecomposition is computed from the resulting images. A reduced-order representation of the object's orientation change is then obtained by projecting the normalized training images into the space spanned by the dominant eigenimages, and interpolating to obtain a manifold.

To determine the pose of the object in a given test image, that image must undergo the same transformations as a training image, i.e., it must be normalized in both scale and intensity and have the average training image subtracted from it. It can then be projected onto the reduced-order eigenspace and the object's orientation can be obtained by computing the closest point on the manifold created using the training images [45]. This process is very computationally efficient and reasonably accurate if the boundary of the object in the test image can be calculated. Unfortunately, the presence of occlusion and/or background clutter complicates this procedure in several ways:

1. The location of the object in the test image cannot be easily determined.
2. Scale normalization cannot be performed on the test image.
3. Brightness normalization is not effective.
4. The occluded and/or cluttered region will alter the projection into the eigenspace.

Some of the problems associated with the pose estimation problem in the presence of background

clutter can be addressed by using a hierarchical eigenspace approach [24], for example, one can include object size variation in the set of training images. To deal with occlusion, one can apply an ''eigen features'' [32], ''attention operator'' [44], or an ''eigen windows'' [33] approach in which small windows around ''feature points'' are automatically detected and used for both training and detection. An alternative is to predefine regions of the image that contain important features and represent them using Gaussian distributions [28]. Unfortunately, these approaches rely on appropriate feature selection as well as detection, and thus lose some of the advantages associated with purely appearance-based techniques.[1]

A search window, which is the ANDed area of the object regions of all images in the training image set, can also be used to deal with the background clutter [29], however, this does not help with occlusion. This search window technique can be extended to an adaptive mask to deal with the occlusion [30], however, occluding objects must be limited to a predefined set. Rather than using fixed masks, Leonardis et al. [34, 37] randomly select pixels from an image in order to identify the most likely object.

Recently, many component-based algorithms have also been proposed for object detection and pose estimation problems in the presence of occlusion and background clutter. In particular, either a product of histograms [38] that represent a wide variety of visual attributes or an adaptive combination of classifiers [39] that are used to classify different components of the object can be used for identification of both ''object'' and ''non-object''. Alternatively, the similarity templates [40], or non-negative matrix factorization technique can be used [41–43] to effectively combine the parts to form a whole. However, some of these approaches lose the advantages associated with purely appearance-based techniques, while others do not result in the minimal orthonormal basis provided by the SVD.

The goal of the work presented here is to solve the pose estimation problem in the presence of occlusion and background clutter, while retaining the framework of an eigendecomposition approach and all its attendant advantages. The next section presents an outline of our approach, which first identifies candidate locations for the object in a test image, and then performs pose estimation using eigendecomposition on a quadtree representation of the training images. The efficacy of our approach, both in terms of accuracy and computational efficiency, as a function of the degree of occlusion and/or background clutter, is explored through a number of experiments.

[1] For purely appearance-based techniques, no modeling is required and thus no feature extraction/selection needs to be performed. Hence these techniques can be applied to any class of objects and can be effectively used in a wide variety of applications [23].

2 Algorithm description

We first consider the problem of object localization and pose estimation under the assumption that the target object is partially occluded but is in an environment where the background can be controlled. A two-step approach is proposed to solve this problem. The first step is to determine the likely candidate locations of the object in the test image. The second step is to evaluate the candidate locations by using eigendecomposition on a quadtree structure of the training images to simultaneously determine if the object is present at a given candidate location, and if so, its pose.

2.1 Localization

Let $\mathbf{y}$ be an image vector of the same size as the training images that represents a window within the test image offset by (v, h) pixels in the vertical and horizontal directions, respectively. Then, if there is no occlusion, one can identify the location of the desired object within the test image, given by (v, h), by comparing $\mathbf{y}$ to the training images for every possible value of (v, h). However, because an eigendecomposition of the training images exists, it is much more computationally efficient to simply compute the amount of $\mathbf{y}$ that can be represented in a smaller eigenspace, i.e.,

$$y^{(k)} = \sqrt{\sum_{i=1}^{k} (\mathbf{u}_i^{\mathrm{T}} \mathbf{y})^2},$$

where k represents the size of the reduced-order representation and $\mathbf{u}_i$ represents the ith eigenimage. Because the brightness within the window $\mathbf{y}$ will vary for different values of (v, h), the normalized measure

$$m_1 = \frac{y^{(k)}}{\|\mathbf{y}\|} \tag{1}$$

is more useful for comparing different locations within the test image. Note that the k-dimensional eigenspace computed for $\hat{X}$ provides the best rank k approximation of $\hat{X}$ [46]. Empirical results for all 16 objects in Fig. 4 showed that each individual image in $\hat{X}$ was also

well represented by its eigenspace but not by the eigenspace computed for any other object. Therefore, measure m_1 in (1) is likely to be maximized when the image represented by $\mathbf{y}$ is similar to one of the training images, thus identifying the location of the object within the test image.

The major computational expense in evaluating (1) consists of the dot products of the eigenimages $\mathbf{u}_i$ with the image vector $\mathbf{y}$ associated with the window at all possible locations, i.e., all values of (v, h). It was shown in [47] that these projections can be efficiently computed by using a 2D FFT. That is, if

$$P = \mathcal{F}^{-1}(\mathcal{F}(\mathcal{X})\mathcal{F}(U_i)^*)$$

then

$$\mathbf{u}_i^{\mathrm{T}}\mathbf{y} = p_{(v,h)},$$

where $\mathcal{X}$ is the test image, U_i is obtained from the eigenimage matrix by padding it with zeros to the size of $\mathcal{X}, \mathcal{F}(\cdot)$ denotes the 2D FFT, * represents the conjugate and $p_{(v,h)}$ represents the (v, h) entry of the matrix P. The 2D FFT of all the eigenimages can be pre-calculated and stored during the off-line process. The major on-line computation involved in evaluating m_1, for every possible location in the test image, requires one 2D FFT of the test image and k 2D inverse FFTs where k is the eigenspace dimension. This is much more efficient than performing a brute force match of the test image with all the training images.

While this method works well for a controlled environment, it is not as effective when occlusion is present. This is illustrated in Fig. 1 where the above approach is applied to the same object, both with and without occlusion. Large values of m_1 do occur when the training images are correctly registered with the test image, however, they do not necessarily correspond to the largest values. In fact, in Fig. 1e there were 202 locations that had a higher or equal value of m_1 than that of the correct location because of the occluding object. It is still possible, however, to differentiate between large values of m_1 that are due to the desired object and those that are due to occlusion. This is due to the likelihood that the value of $y^{(k)}$ will be much more sensitive to small registration errors for the desired object than for the occluding object. This motivates the use of a measure based on the second derivative of $y^{(k)}$, namely,

$$m_2 = \sqrt{\left(\frac{\partial^2 y^{(k)}}{\partial v^2}\right)^2 + \left(\frac{\partial^2 y^{(k)}}{\partial h^2}\right)^2}. \quad (2)$$

This measure uses the fact that eigenspace approaches are highly sensitive to registration errors [48], i.e., measure m_1 will have a large value if the center of the object in the test window is perfectly aligned with the center of its training images; even a small registration error will result in a much smaller value of m_1. Thus, measure m_2 will typically have a large value only when the test window is perfectly registered with the training images. This behavior is not likely if the eigenspace was not computed for the object in the test window. The example in Fig. 1 shows that this measure is clearly effective in identifying the correct object location.

Because the measure m_2 can tend to be "noisy" due to the use of derivatives, it is combined with the value-based measure m_1 to form the measure:

$$M = \begin{cases} m_2 & \text{if } m_1 \geq \rho, \\ 0 & \text{if } m_1 < \rho, \end{cases}$$

where ρ is a preset threshold, which is used to identify candidate locations of the desired object even under the influence of occlusion.

An experiment was conducted to evaluate the accuracy of measure M for use in identifying the location of a desired object in a test image as a function of the percent of occlusion. (The percent of occlusion for a test image is defined as the area of the object that is occluded divided by the area of the entire object.) A total of 800 cases were examined, with the percent of occlusion evenly distributed between 0 and 80%. The target object used in this experiment is shown in Fig. 1a with the image of an occluding object randomly selected from a pool of 15 other objects (see Fig. 4) to create the desired level of occlusion. The test images were of size 256×256 and the training images were of size 128×128, with a total of 90 training images used to create a 12-dimensional eigenspace.

The measure M was evaluated at a resolution of one pixel in both horizontal and vertical directions. The number of locations that have a measure higher than or equal to that of the correct location, will be referred to as the rank of the correct location. In 60% of all the cases, the rank of the correct location was one, i.e., it had the highest value of M. In addition, the rank of the correct location was less than fifty for over 90% of all cases. The average rank was never more than 100, even for the maximum occlusion of 80%. It is important to note that a larger rank value results in a longer computation time, but does not affect solution accuracy. This suggests that an object registration and pose estimation scheme based on candidate locations identified using

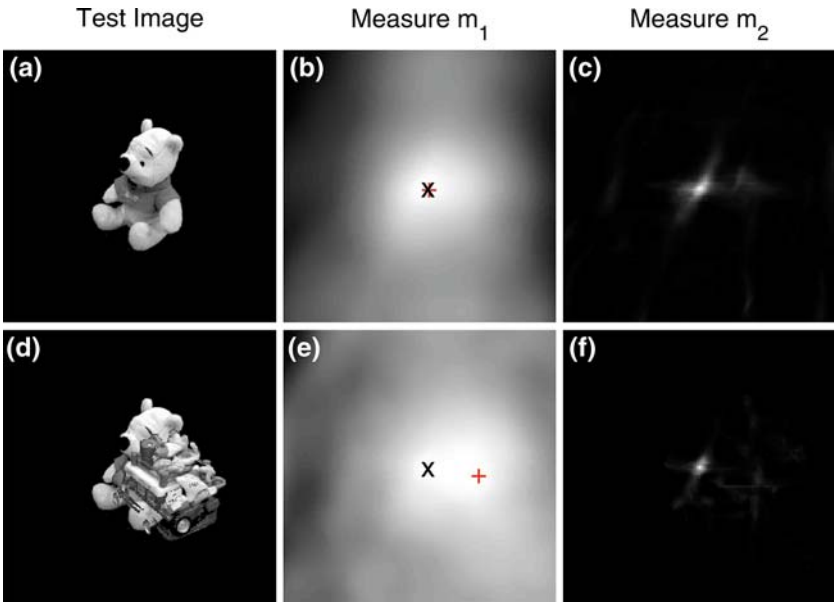

Fig. 1 Test images without and with occlusion are shown in (**a**) and (**d**), respectively. Their values of measure m_1 are shown in (**b**) and (**e**) with the peak location of the measure marked by a "+" and the correct location of the object marked by an "×". Note that the peak has shifted from the correct location due to the occlusion. However, the peak for measure m_2, shown in (**c**) and (**f**), correctly registered the location of the object even when occlusion is present

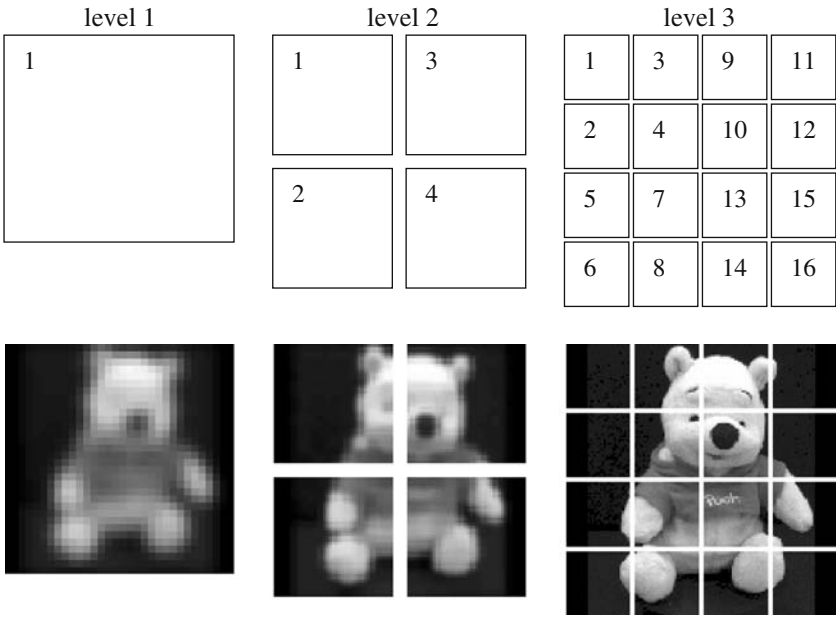

Fig. 2 This figure shows how an image is decomposed into a quadtree structure, and how the sub-images in each level are numbered

the measure M may be very efficient.[2] This is the topic of the next subsection.

2.2 Quadtree-based detection

Once a number of candidate locations have been determined using the measure M, these locations are evaluated using eigendecomposition on a quadtree representation of the training images. Eigenimages are calculated for each level of a quadtree decomposition of the training images. At a level l, each training image is broken into $4^{(l-1)}$ sub-images (see Fig. 2). Let $\mathbf{x}_{i,l,j}$ denote the image vector associated with the jth sub-image in level l of the ith training image. Then the image data matrix associated with the jth sub-image in level l is formed as

$$X_{l,j} = [\mathbf{x}_{1,l,j}\ \mathbf{x}_{2,l,j}\ \dot{s}\ \mathbf{x}_{n,l,j}].$$

Eigendecomposition is applied to $\hat{X}_{l,j} = X_{l,j} - \bar{X}_{l,j}$ for each sub-image in each level where $\bar{X}_{l,j}$ is the average

[2] Note that when the actual object location is not of rank one, the rank one candidate is frequently far from the correct location (due to occlusion) so that local optimization techniques such as gradient descent [32] are not effective.

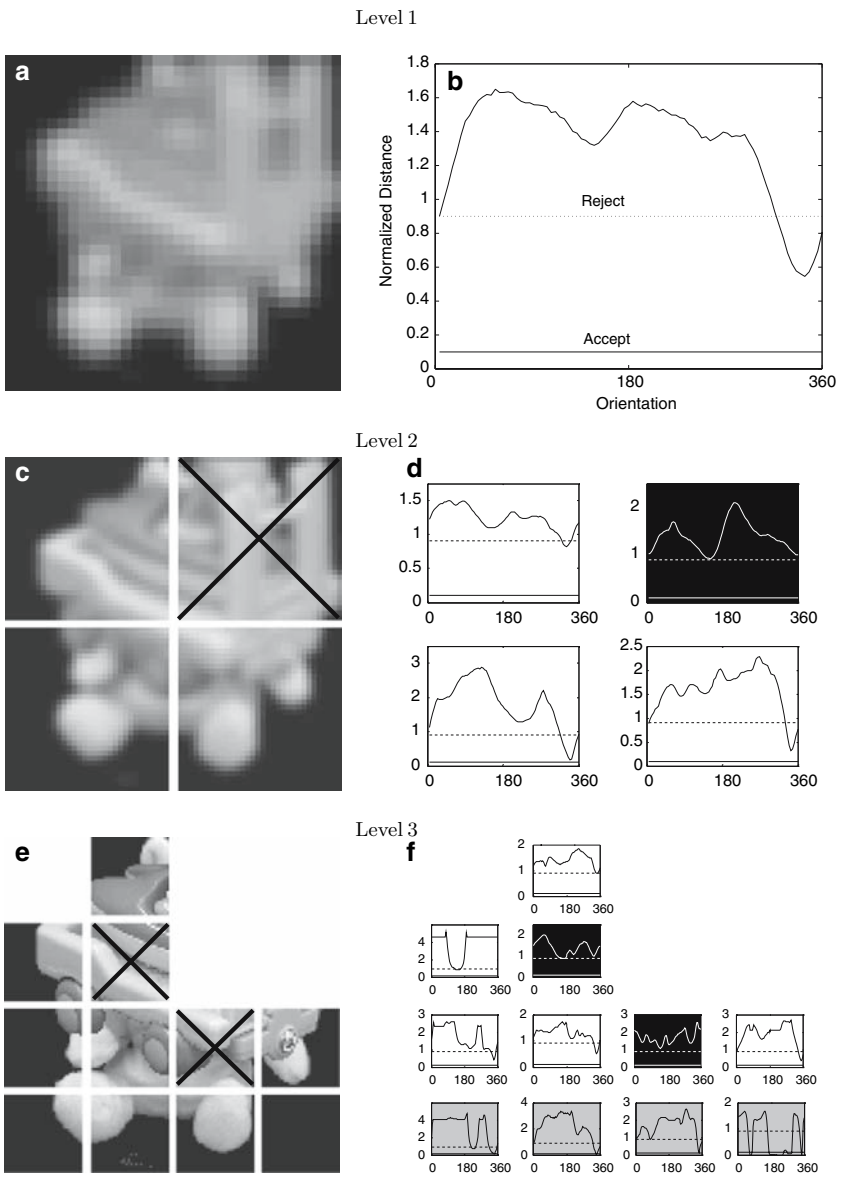

Fig. 3 This figure shows an example of the quadtree eigenspace object/pose estimation process. The sub-figures on the left show the test image and the result of the process for all sub-images in each level. A *box with a cross* through it indicates the rejected sub-images. The sub-figures on the right show the normalized distance as a function of orientation, for each sub-image in each level. The acceptance and rejection thresholds are set at 0.1 and 0.9, respectively. A *sub-plot with a gray background* indicates that the normalized distance to some of the training images went below the acceptance threshold for the corresponding sub-image, while a *sub-plot with a black background* indicates that the normalized distance to all the training images was above the rejection threshold for the corresponding sub-image

image data matrix for the image vectors in $X_{l,j}$. If a k-dimensional eigenspace is used, then there will be $4^{(l-1)} \times k$ eigenimages for this level. The image data matrices $X_{l,j}$ that contain very little information about an object, i.e., contain mostly background, are automatically discarded by the algorithm (for example, $X_{3,1}$ in Fig. 2).[3] The projection of $\mathbf{x}_{i,l,j} - \bar{\mathbf{x}}_{l,j}$ onto the corresponding eigenspace is calculated to form the pose manifold that is used in the on-line process.

The on-line process consists of performing image comparisons in the eigenspace for each of the candidate locations until a "match" is found. For pose estimation without occlusion, the orientation of the object is obtained from the point on the manifold that is closest to the projection of the test image because both the test image and training images can be normalized. When occlusion is present, the images are not automatically registered so that the distance to the manifold of training images is used to simultaneously determine if the object is present at this location in addition to determining its orientation. The following normalized distance is used:

$$d = \|\mathbf{t}_\theta - \mathbf{p}_q\| / \|\mathbf{t}_\theta\|, \tag{3}$$

where $\mathbf{t}_\theta$ represents a point on the manifold of training images at orientation θ for a particular sub-image and $\mathbf{p}_q$ represents the eigenspace projection of the

[3] Specifically, the image data matrices corresponding to the training sub-images, whose rank is below 12, are automatically discarded.

Fig. 4 The 16 target objects used for the experiments in this section. For each target object, 90 training images were obtained by rotating the object by 4° between images. All training images are of size 128×128 pixels with 8 bits used to represent intensity

corresponding sub-image in the window associated with the qth candidate location. The projection $\mathbf{p}_q$ is considered to match $\mathbf{t}_i$, the ith training image whose orientation is closest to θ, when d is less than a preset threshold. When the training images cover all the possible variations in orientation, illumination, scaling and other factors, this threshold can be set very small; however this is usually not feasible and interpolation is used between samples. (A threshold of 0.1 was used for the examples presented in the subsequent sections, which worked well for a variety of objects.)

For each candidate location, the image comparison in the eigenspace is evaluated based on (3). The comparison starts at the first level and proceeds to smaller images at higher levels in the quadtree. At each level, all the sub-images that do not consist of only the background are examined. If d is smaller than a preset threshold for orientation θ, the orientation associated with training image i whose projection is closest to $\mathbf{t}_\theta$ will receive a vote. The value of this vote is equal to the percentage of the corresponding training sub-image that is occupied by the object. For example, the vote associated with sub-image 4 at level 3 in Fig. 2 is much smaller than that of sub-image 7 at level 3. This mechanism is used to de-emphasize the vote from a sub-image corresponding to a background area or containing very limited information about the object. If the normalized distances for all orientations of a sub-image are greater than a preset threshold (0.9 is used), all child notes in the following level will be skipped to save computation time. If the vote for a particular orientation exceeds a preset threshold, the process is terminated and the orientation of the object is determined by the orientation receiving the largest vote. The process moves to the next level if the maximum vote does not exceed the preset threshold. This threshold is set to $\frac{2}{4^{(l_{\max}-l)}}$where $l_{\max}$ is the maximum level allowed. (In this work $l_{\max} = 4$ is used with a training image of size 128×128.) If the level $l_{\max}$ is completed and no voting exceeds this threshold, the next candidate location is then assessed.

An example of this process is illustrated in Fig. 3. The algorithm starts from the first level, with the corresponding area of the test image displayed on the left

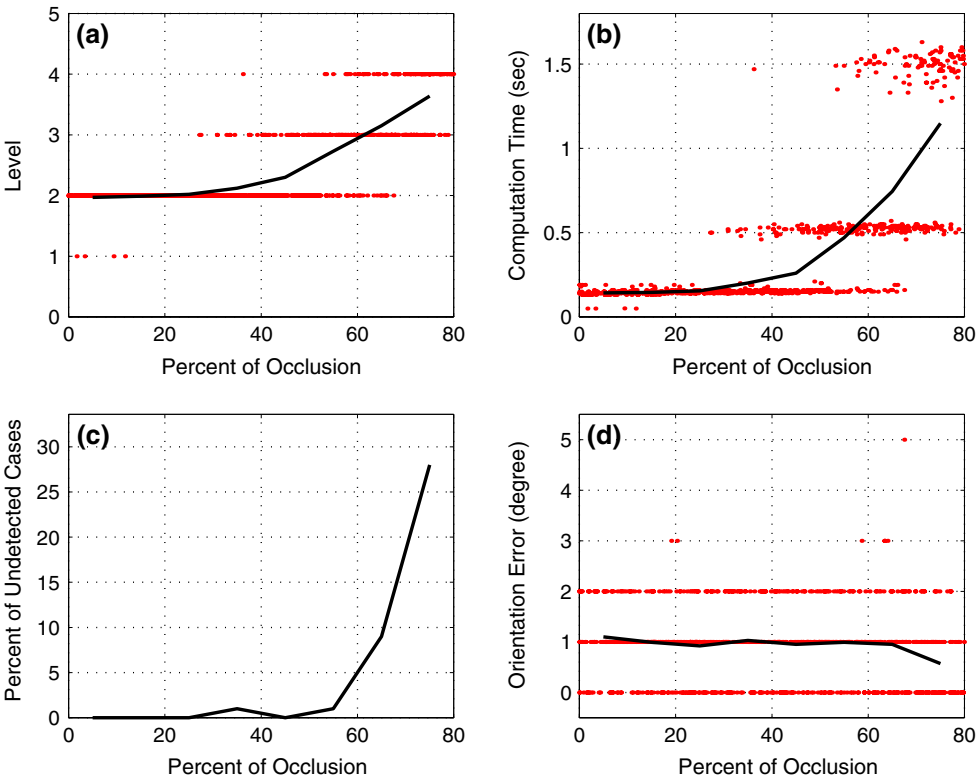

Fig. 5 This figure provides data on the accuracy and computational efficiency of the proposed algorithm as a function of the percent occlusion when evaluated on 800 random test cases of correctly registered images: **a** Depth to which the quadtree is searched (average plotted with a *solid line*), **b** computation time required for each case (average plotted with a *solid line*), **c** number of cases where the pose cannot be determined, **d** orientation error when a pose is determined (average plotted with a *solid line*)

and the normalized distance for different orientations on the right. The acceptance threshold is set at 0.1 and the rejection threshold is set at 0.9. In this example, neither the acceptance nor the rejection criterion is satisfied at the first level; therefore, the search proceeds to the second level. In the second level, the third sub-image exceeded the rejection threshold, and thus none of its child nodes are evaluated. In level 3, sub-images 6, 8, 14, and 16 were not occluded and thus were successfully detected. All four of these sub-images voted for an orientation of 355°, although the last sub-image also voted for other orientations. However, because the value of a vote is based on the percent of the matched training sub-image that is occupied by the object, the value of the vote for this sub-image is equal to zero. The process stops at this level with the conclusion that the object is at an orientation of 355°.

3 Experimental evaluation

To evaluate the accuracy and computational efficiency of the proposed algorithm, a number of experiments were performed. For these experiments, it was assumed that the target object whose pose is desired is partially occluded but it is located in a controlled environment with no background clutter. A variety of objects (see Fig. 4) were used to test the robustness of the proposed algorithm. A 12-dimensional eigenspace was used for every sub-image at each level of the quadtree.[4] All test images (of size 256×256) were generated by superimposing the image of a randomly selected occluding object on top of a randomly selected target object.[5] The target objects were selected from 360 different possible images, i.e., a 1° rotation between successive images, in order to include poses that were not part of the training set.[6] The percent of occlusion used in the test images was equally distributed, with 100 cases selected within each 10% range. Finally, the training data was used to do a receiver operating characteristics (ROC) analysis to find the tradeoff between false positives and true negatives for different thresholds. Using this analysis, the thresholds that gave the optimum results were used in the

[4] Empirical results showed that using a constant subspace dimension at every sub-image performs consistently better than using a constant energy recovery ratio. The main reason behind this is that a constant subspace dimension tends to make the energy recovery ratio increase as the algorithm searches further down the quadtree.

[5] The generation of the occluded test images in this manner can induce artifacts, like large step edges along the boundaries, however, our results indicate that these artifacts do not affect the performance of the algorithm.

[6] We elected not to use one of the "standard" object data sets, like COIL-100 [49], COIL-10 [50], SOIL-47 [51], and ALOI [52], because they only contain 72 orientations per object.

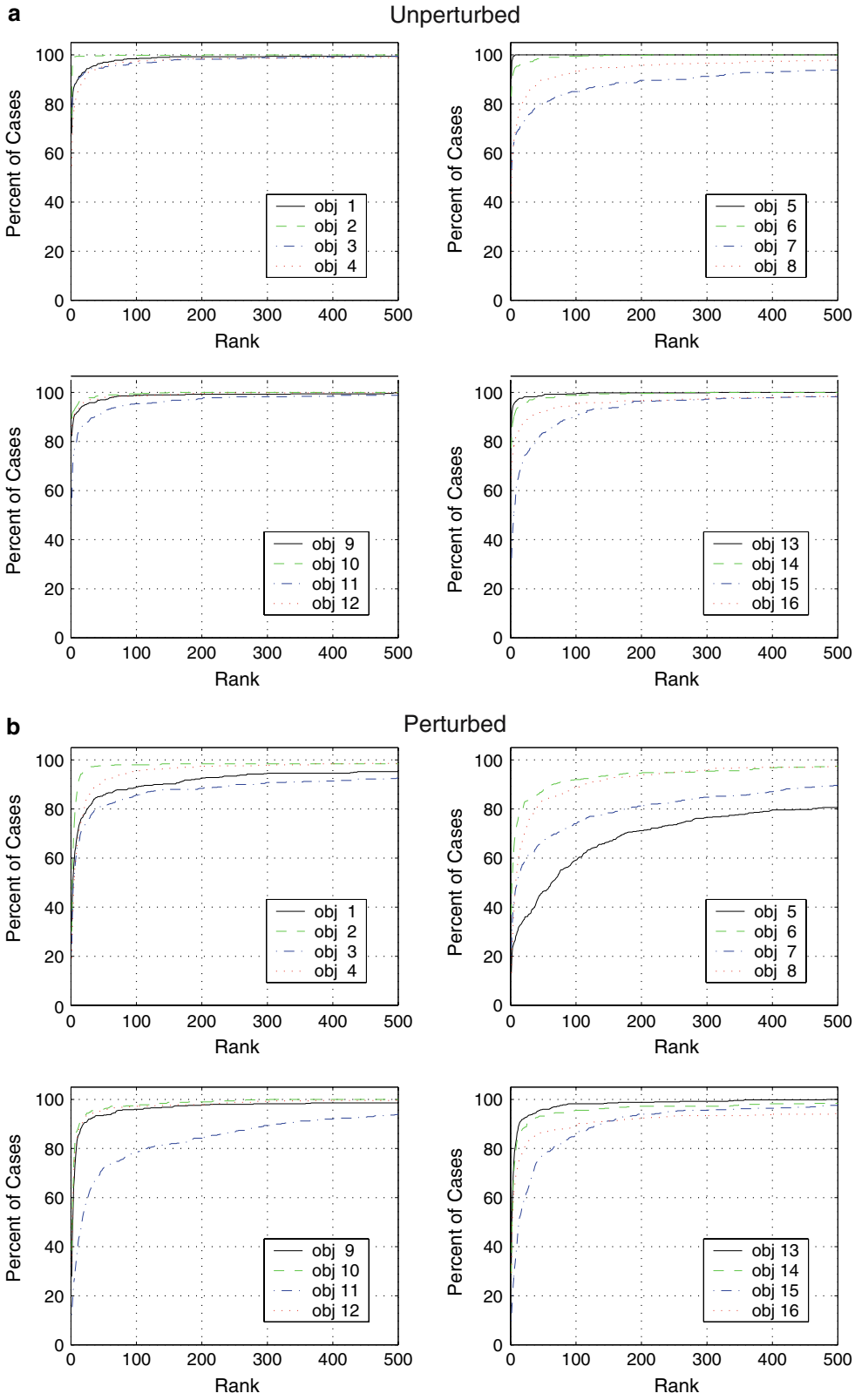

Fig. 6 Performance of the proposed algorithm when using the measure M to select candidate locations. The percent of cases that have the rank of the correct location smaller than the value of the x-axis is displayed

Table 1 Algorithm performance on 500 UNPERTURBED test images (all numbers are percentages)

Object	Unidentified	Location errors	Orientation errors	Total errors	Total correct
1	0.2	0.6	0.6	1.2	98.6
2	0.2	0.0	0.0	0.0	99.8
3	0.8	1.8	1.4	3.2	96.0
4	1.6	2.0	0.8	2.8	95.6
5	1.6	0.0	0.4	0.4	98.0
6	7.2	0.0	0.0	0.0	92.8
7	12.4	1.4	1.4	2.8	84.8
8	1.2	7.8	1.2	9.0	89.8
9	0.6	0.2	0.0	0.2	99.2
10	0.8	0.0	0.4	0.4	98.8
11	7.0	0.0	0.0	0.0	93.0
12	0.4	1.0	1.0	2.0	97.6
13	0.0	0.4	0.8	1.2	98.8
14	0.2	0.0	1.2	1.2	98.6
15	0.8	3.8	0.2	4.0	95.2
16	4.8	0.0	0.0	0.0	95.2
Best case	0.0	0.0	0.0	0.0	99.8
Average case	0.8	0.3	0.5	1.2	96.8
Worst case	12.4	7.8	1.4	9.0	84.8

Table 2 Algorithm performance on 500 PERTURBED test images (all numbers are percentages)

Object	Unidentified	Location errors	Orientation errors	Total errors	Total correct
1	1.4	4.4	1.4	5.8	92.8
2	1.2	0.0	0.0	0.0	98.8
3	1.4	5.6	5.4	11.0	87.6
4	1.4	3.8	5.2	9.0	89.6
5	17.4	1.6	1.6	3.2	79.4
6	10.0	0.2	0.6	0.8	89.2
7	12.0	4.8	3.8	8.4	79.6
8	1.0	15.6	1.8	17.4	81.6
9	1.6	0.0	1.8	1.8	96.6
10	2.2	0.0	3.0	3.0	94.8
11	12.0	0.4	0.2	0.6	87.4
12	1.2	1.2	1.2	2.4	96.4
13	0.2	1.0	0.8	1.8	98.0
14	1.2	1.2	1.8	3.0	95.8
15	0.2	7.2	1.2	8.4	91.4
16	6.6	0.4	0.2	0.6	92.8
Best case	0.2	0.0	0.0	0.0	98.8
Average case	1.4	1.2	1.5	3.0	92.1
Worst case	17.4	15.6	5.4	17.4	79.4

proposed algorithm. In particular, the rejection threshold was set at 0.9 and the acceptance threshold was set at 0.1.

3.1 Quadtree approach evaluation

The first set of tests was designed to evaluate the performance of the quadtree approach, independent of the localization problem. For these tests, the size of the target object was the same as for the training images and the objects location in the test image was specified. Figure 5 shows the results of these tests where the percent of occlusion was varied from 0 to 80%, i.e., 800 test cases. As would be expected, the amount of work that the algorithm must perform, i.e., the depth to which the quadtree must be evaluated, is monotonically related to the difficulty of the problem, i.e., the percent occlusion. Both the average depth and average computation time markedly increase for objects that are occluded by more than 50%. The difficulty of determining the pose of objects that are more than 50% occluded is even more strikingly evident in part (c) of the figure which plots the number of cases in which the algorithm cannot determine the objects orientation. However, it is important to note that even for the 100 cases with occlusions between 70–80%, in 72 of them the pose of the object was able to be determined. Even more importantly, the average accuracy to which the algorithm determines an object's orientation is essentially independent of the amount of occlusion. In other words, if the algorithm can make a decision regarding an object's pose, it is usually quite accurate.

3.2 Pose estimation with occlusion

The next set of experiments was designed to evaluate the performance of the quadtree decomposition approach when applied to candidate object locations identified by using measure M on the test images. To account for the fact that size normalization cannot be performed, an additional 180 training images were used for each object, where the size of the object was enlarged and reduced by 5%, resulting in an image data matrix of 270 images. Two sets of test images were generated in the manner described above. In one set, referred to as the perturbed set, the object size, the brightness of the background and the brightness of the object itself were all randomly perturbed by a value between 0 and 5%. In the other set all of these factors were held constant (referred to as the unperturbed set). Because the quadtree-based pose estimation approach started to degrade when the occlusion was greater than 50%, 500 test cases were used for both the perturbed and unperturbed sets with the percent occlusion equally distributed between 0 and 50%. The measure M was used to select the top 100 candidate locations with the constraint that two candidate locations cannot be adjacent.

The performance of the measure M for localization is shown in Fig. 6, where the percent of cases that have the rank of the correct location smaller than the value

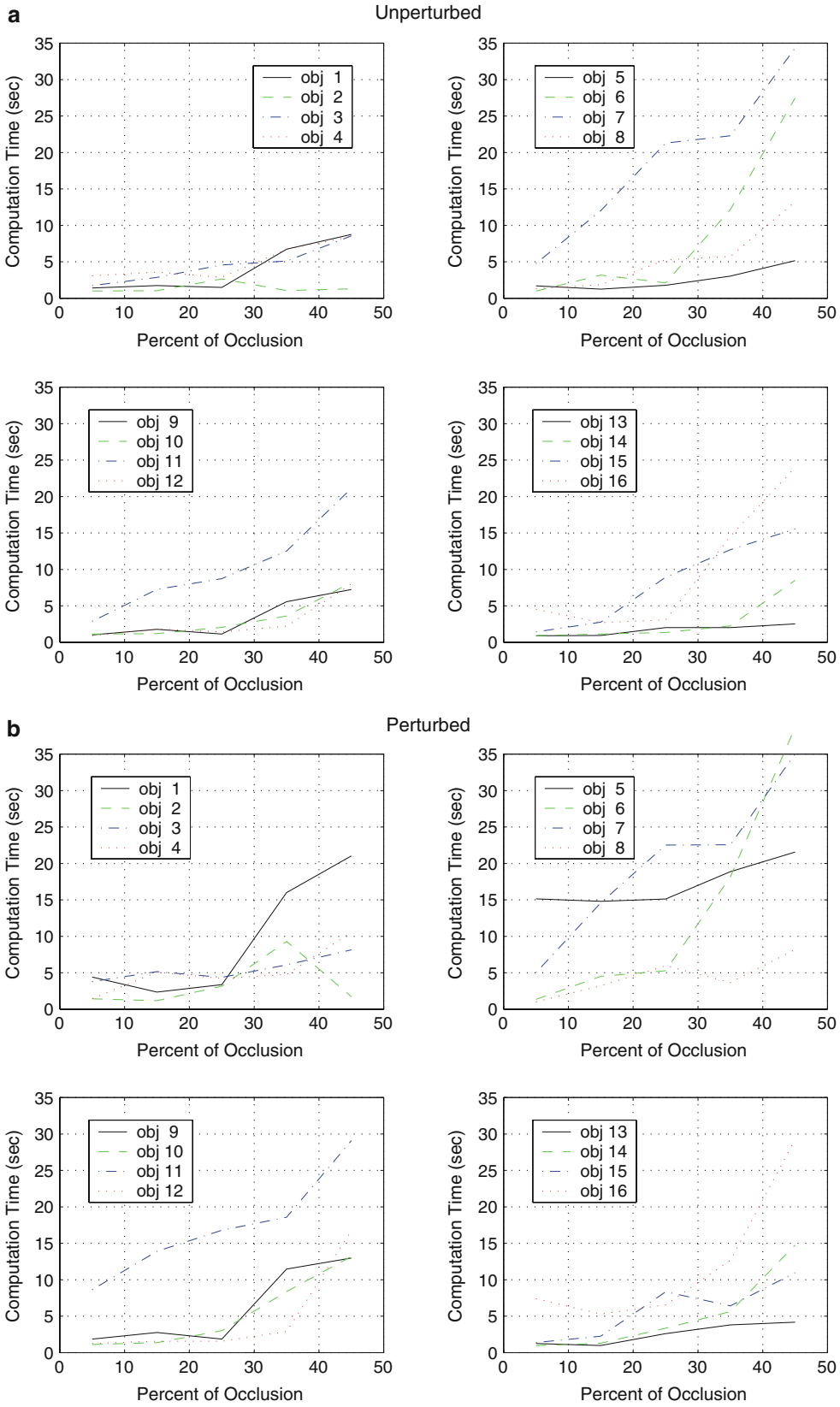

Fig. 7 Average computation time as a function of percent of occlusion. (All programs are written in MATLAB and executed on an HP9000/C110 workstation.) This result is for the experiment in Sect. 3

Table 3 Comparison of the proposed algorithm versus the eigen window approach on 500 unperturbed test images

Object	Percent occlusion	Proposed algorithm			Eigen window	
		Unidentified	Location errors	Total errors	Location errors	Total errors
2	10	0	0	5	11	26
	20	0	0	6	10	25
	30	0	0	2	17	28
	40	0	0	4	16	29
	50	0	0	8	20	30
	Total	0	0	25	74	138
7	10	0	5	18	15	21
	20	0	18	27	34	43
	30	3	18	29	40	47
	40	0	24	29	48	56
	50	10	29	35	47	48
	Total	13	94	138	184	215
12	10	0	0	7	33	36
	20	0	0	2	32	37
	30	0	2	6	32	39
	40	0	1	6	30	42
	50	1	8	17	41	48
	Total	1	11	38	168	202
4	10	0	4	16	31	51
	20	0	6	15	42	56
	30	0	4	16	45	53
	40	0	9	27	51	68
	50	3	19	35	61	69
	Total	3	42	109	230	297

Fig. 8 One frame of a video sequence used to evaluate the proposed algorithm. These images were not occluded, but contained background clutter. The quadtree eigenspace algorithm was used to identify the left-most ship in this image and to determine its pose

of the x-axis is displayed. The rank displayed here is before the adjacent pixels were removed, so that the locations with the top 200 values of M are candidate locations that will be evaluated by the quadtree approach. Thus for all objects in the unperturbed set, 90% of the cases will have the correct location of the object evaluated. This percentage is reduced significantly if perturbation in the test images is allowed. In particular, in the worst case, i.e., object five, the correct location was a candidate in only 70% of the perturbed cases. This is due to object five being rotationally symmetric and therefore having its eigenimages contain sharp edges that are very sensitive to size variations. This situation can be addressed by including more size variation into the training set. (Note that the correct location for object five had a rank of less than 10 for all cases in the unperturbed set.)

The performance of the complete algorithm that includes the quadtree detection applied to the candidate locations identified using the measure M is summarized in Tables 1 and 2 for unperturbed and perturbed test sets, respectively. Unidentified cases refer to those test cases where the pose estimation procedure did not identify the object at any of the candidate locations. This is either due to the correct location not being one of the candidate locations (which was typically the case for objects 5, 7, and 11) or due to a significant difference in the appearance of the test image (either due to a pose that is not represented

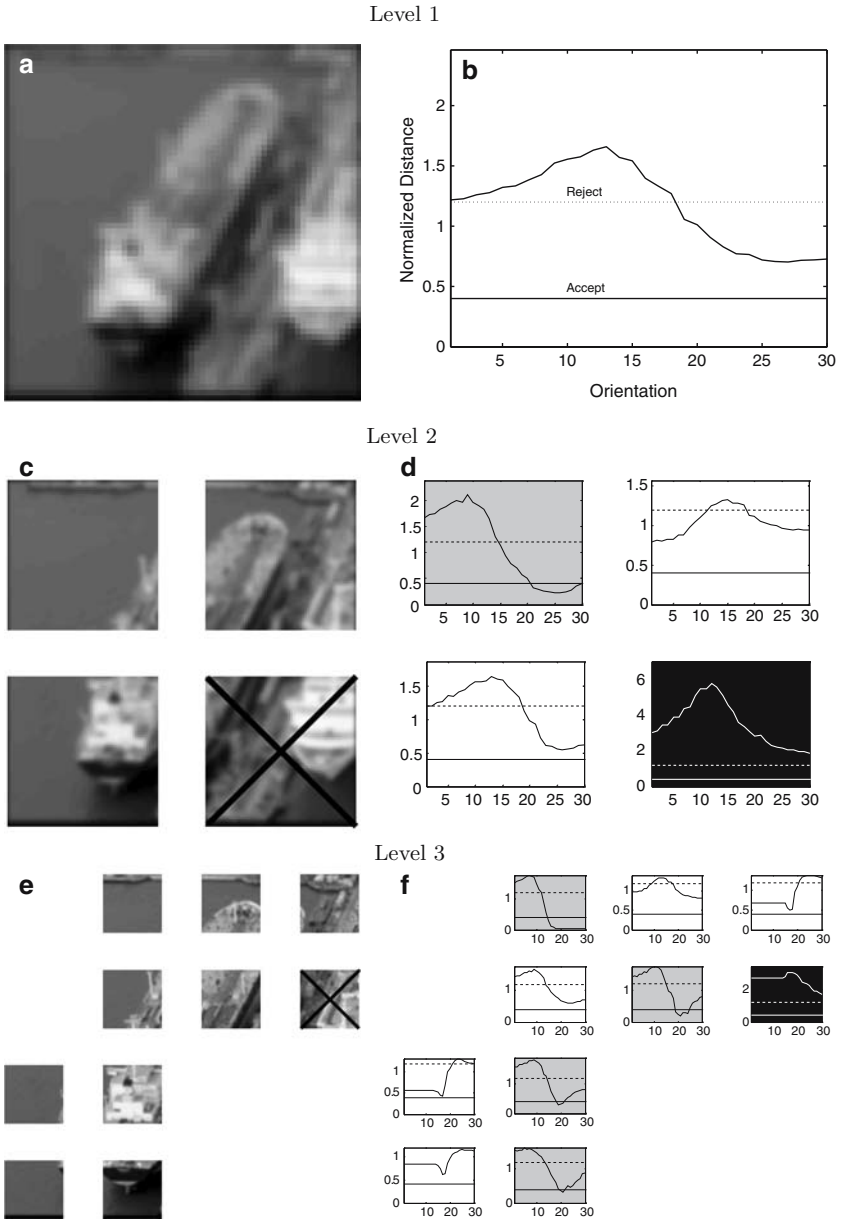

Fig. 9 This figure shows an example of the quadtree eigenspace object/pose estimation process for the rank 1 candidate location from the frame given in Fig. 8. The *sub-figures on the left* show the test window and the result of the process for all sub-images in each level. A *box with a cross* through it indicates a rejected sub-image. The *sub-figures on the right* show the normalized distance as a function of orientation, for each sub-image in each level. The acceptance and rejection thresholds are set at 0.4 and 1.2, respectively. A *sub-plot with a gray background* indicates that the normalized distance to some of the training images went below the acceptance threshold for the corresponding sub-image, while a sub-plot with a black background indicates that the normalized distance to all the training images was above the rejection threshold for the corresponding sub-image

in the training images, occlusion, or perturbation) that prevents detection even at the correct location (which was typically the case for objects 6 and 16). Location errors refer to those cases where the algorithm identified the presence of the object but in an incorrect location. This typically occurred for objects that either contain large areas of uniform intensity (objects 3, 7, 8 and 15) or have different portions of the object that appear similar (objects 1 and 4). For those cases where the object was identified in its correct location, the error in computing the object's pose was calculated. Note that because the training images are taken every 4°, an error of up to 4° still implies that the algorithm identified the correct interval in the manifold of object poses. (Thus a more accurate orientation could be determined by doing a local optimization.) Therefore, only orientation errors of greater than 8° are considered pose estimation errors. The total number of errors is the sum of the cases where either the location or the pose was incorrectly determined.

In general, the percentage of cases where the algorithm correctly identified both the location and the pose of the target object was quite high. In particular, most objects were correctly identified (95% in the unperturbed case and 90% in the perturbed case), or if they were not, then the algorithm effectively declared that the problem was too difficult, i.e., it could not register the test image (e.g., for objects 5, 6, 7, 11, and 16). The only objects that created a difficulty for the algorithm in terms of true errors were object 8 and, to a

lesser extent, objects 3 and 4 for the perturbed case. (The localization of object 8 is inherently difficult due to the large areas of uniform appearance that are present at many different poses.) The amount of work that the algorithm performs, i.e., the computation time, is directly related to the difficulty of the detection problem (see Fig. 7). In general, objects that are more difficult to localize (objects 5, 7, and 11 in the perturbed set) require the most computation time.

To provide a relative measure of performance, the proposed algorithm was compared to the eigen window approach [33]. This approach was selected because it was similar in that it made decisions based on sub-windows within the image. However, because it is a computationally expensive approach, the number of orientations used in the training set was reduced. In particular, 36 equally spaced images of an object were used for obtaining the training manifold. Objects 2, 7, 12, and 4 from Fig. 4, which gave the maximum, minimum, median, and mean accuracy rates, respectively, were used as representative examples. Table 3 shows the results for the test cases corresponding to 10% to 50% occlusion. Because the eigen window approach always assumes to have found the object in the scene, there is no ''unidentified'' column for that approach in the table. The proposed algorithm outperformed the eigen window approach in all cases, primarily due to the fact that the eigen window approach had more difficulty in locating the object correctly.

3.3 Pose estimation with background clutter

3.3.1 Without occlusion

The quadtree eigenspace object/pose estimation process was applied to a video sequence with background clutter, but without occlusion. Figure 8 shows one frame from the sequence.[7] The objective was to identify the smaller ship on the left and to determine its pose. The training process was performed on images that only included the desired ship superimposed on a black background. These training images had a variability in the viewing angle of approximately 45° for the camera as it panned past the ships. The algorithm was then tested on the entire video sequence. A search window, which contained the ANDed area [29] of the object regions of all training images, was used to mask out the background clutter in the testing sub-images. In 88.57% of the cases, our localization procedure identified the desired ship with rank 1 and in the remaining cases, it identified a ship of the same size and form. Figure 9 shows the results of the quadtree approach applied to the rank 1 candidate location from the frame given in Fig. 8. Even in the presence of background clutter, our procedure detected the correct orientation at level 3.

3.3.2 With occlusion

The quadtree eigenspace pose estimation process was also applied to partially occluded objects against cluttered backgrounds [53]. For this evaluation, occluded objects from Fig. 4 were placed in an environment with background clutter. During the online process, a search window [29] was again used to mask out the background clutter in the test sub-image for the candidate location under consideration. An acceptance threshold of 0.35 was used on a list of 1,000 candidate locations for both perturbed and unperturbed cases.

Clearly, the addition of background clutter makes the object detection/pose estimation problem much more difficult and so the accuracy rates go down. However, the different stages in the algorithm are not affected uniformly. In particular, the addition of background clutter did not appreciably degrade the object detection accuracy. The effect on object localization was variable. While the median increase in the number of location errors was only 14%, two object were incorrectly located half of the time. The objects that were difficult to localize had either (1) a small object area but a large AND area or (2) large internal areas of uniform intensity. The first case is difficult because large areas of the background will be included in the localization process. The second case is problematic because background clutter tends to contaminate edge information and there is little information in the interior. If an object is localized properly, the background clutter has little effect on pose estimation, increasing the average orientation error by only 1.8%.

4 Conclusion

This paper has presented an algorithm based on applying eigenspace methods to a quadtree representation of a set of related images to solve the pose estimation problem in the presence of occlusion and/or background clutter. Because the algorithm relies purely on the appearance of the objects in the training set of images, it is very general and easy to apply. The

[7] A video sequence of ship images with resolution of 720 × 1,280 pixels each was provided by the National Imagery and Mapping Agency.

difficulties that are created due to the presence of occlusion and/or background clutter, i.e., the inability to easily locate the desired object and apply the appropriate normalizations, are efficiently overcome by the recursive quadtree procedure. While on-line detection times can be an order of magnitude larger than for unoccluded images, the amount of work is proportional to the difficulty of the problem, i.e., the extent of the occlusion. In addition, the algorithm rarely makes an error in detecting the location and pose of the desired object, preferring to declare the detection problem too difficult when too much information is occluded.

4.1 Originality and contribution

Purely appearance-based techniques such as SVD have been extensively used in many computer vision applications, i.e., face characterization, object recognition, pose estimation, visual tracking, and inspection. The fundamental problem of object recognition and pose estimation of three-dimensional objects is considered here. Unfortunately, one of the drawbacks associated with the appearance-based techniques for solving this problem is that they are very sensitive to occlusion and background clutter. Various different approaches have been proposed until now to solve this problem. However, some of these approaches, e.g., feature-based algorithms, lose the advantages associated with purely appearance-based techniques. The work in this paper presents an algorithm that is based on applying the SVD to a quadtree representation of the image dataset used to describe the appearance of an object. This allows decisions concerning the pose of an object to be based on only those portions of the image in which the algorithm has determined that the object is not occluded. The novelty of this algorithm lies in the combination of the SVD with the quadtree decomposition. This combination allows one to solve the pose estimation problem in the presence of occlusion and background clutter, while retaining the framework of an eigendecomposition approach and all its attendant advantages. The empirical results show that this computationally efficient algorithm simultaneously recognizes the object and its pose in the test scene with very good accuracy.[8]

[8] The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the US Government.

Acknowledgments This work was supported in part by the Office of Naval Research under contract no. N00014-97-1-0640, the National Imagery and Mapping Agency under contract no. NMA201-00-1-1003, through collaborative participation in the Robotics Consortium sponsored by the US Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0012, and the Missile Defense Agency under the contract no. HQ0006-05-C-0035. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. A preliminary version of this work was presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems held at Maui, Hawaii, October 29–November 3, 2001.

References

1. Fukunaga K (1990) Introduction to Statistical Pattern Recognition, 2nd edn. Academic, London
2. Martinez AM, Kak AC (2001) PCA versus LDA. IEEE Trans PAMI 23(2):228–233
3. Sirovich L, Kirby M (1987) Low-dimensional procedure for the characterization of human faces. J Opt Soc Am 4(3):519–524
4. Kirby M, Sirovich L (1990) Application of the Karhunen–Loeve procedure for the characterization of human faces. IEEE Trans PAMI 12(1):103–108
5. Turk M, Pentland A (1991) Eigenfaces for recognition. J Cogn Neurosci 3(1):71–86
6. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Trans PAMI 19(7):711–720
7. Brunelli R, Poggio T (1993) Face recognition: Features versus templates. IEEE Trans PAMI 15(10):1042–1052
8. Pentland A, Moghaddam B, Starner T (1994) View-based and modular eigenspaces for face recognition. In: Proceedings of IEEE conference computer vision and pattern recognition. Seattle, WA, pp 84–91
9. Yang MH, Kriegman DJ, Ahuja N (2002) Detecting faces in images: A survey. IEEE Trans PAMI 24(1):34–58
10. Murase H, Sakai R (1996) Moving object recognition in eigenspace representation: Gait analysis and lip reading. Pattern Recogn Lett 17(2):155–162
11. Chiou G, Hwang J-N (1997) Lipreading from color video. IEEE Trans Image Process 6(8):1192–1195
12. Murase H, Nayar SK (1994) Illumination planning for object recognition using parametric eigenspaces. IEEE Trans PAMI 16(12):1219–1227
13. Huang CY, Camps OI, Kanungo T (1997) Object recognition using appearance-based parts and relations. In: Proceedings of IEEE conference on computer vision and pattern recognition. San Juan, PR, USA, pp 877–883
14. Campbell RJ, Flynn PJ (1999) Eigenshapes for 3D object recognition in range data. In: Proceedings of IEEE conference on computer vision and pattern recognition. Fort Collins, CO, USA, pp 505–510
15. Jogan M, Leonardis A (2000) Robust localization using eigenspace of spinning-images. In: Proceedings of IEEE workshop omnidirectional vision. Hilton Head Island, South Carolina, USA, pp 37–44
16. Borgefors G (1988) Hierarchical chamfer matching: A parametric edge matching algorithm. IEEE Trans PAMI 10(6):849–865
17. Yoshimura S, Kanade T (1994) Fast template matching based on the normalized correlation by using multiresolution

eigenimages. In: 1994 IEEE workshop motion of non-rigid and articulated objects, Austin, Texas, pp 83–88
18. Winkeler J, Manjunath BS, Chandrasekaran S (1999) Subset selection for active object recognition. In: Proceedings of IEEE conference computer vision and pattern recognition. Fort Collins, Colorado, USA, pp 511–516
19. Martinez AM, Vitria J (2001) Clustering in image space for place recognition and visual annotations for human–robot interaction. IEEE Trans Syst Man Cybern 31(5):669–682
20. Crowley JL, Pourraz F (2001) Continuity properties of the appearance manifold for mobile robot position estimation. Image Vis Comput 19(11):741–752
21. Nayar SK, Murase H, Nene SA (1994) Learning, positioning, and tracking visual appearance. In: Proceedings of IEEE international conference on robotics and automation, San Diego, CA, USA, pp 3237–3246
22. Black MJ, Jepson AD (1998) Eigentracking: robust matching and tracking of articulated objects using a view-based representation. Int J Comput Vis 26(1):63–84
23. Murase H, Nayar SK (1995) Visual learning and recognition of 3-D objects from appearance. Int J Comput Vis 14(1):5–24
24. Murase H, Nayar SK (1997) Detection of 3D objects in cluttered scenes using hierarchical eigenspace. Pattern Recogn Lett 18(4):375–384
25. Nayar SK, Nene SA, Murase H (1996) Subspace method for robot vision. IEEE Trans Rob Autom 12(5):750–758
26. Moghaddam B, Pentland A (1997) Probabilistic visual learning for object representation. IEEE Trans PAMI 19(7):696–710
27. Chang C-Y, Maciejewski AA, Balakrishnan V (2000) Fast eigenspace decomposition of correlated images. IEEE Trans Image Process 9(11):1937–1949
28. Martinez AM (2002) Recongnizing imprecisely localized, partially occluded, and expression varient faces from a single sample per class. IEEE Trans PAMI 24(6):748–763
29. Nayar SK, Murase H (1995) Image spotting of 3D objects using parametric eigenspace representation. In: Proceedings of 9th Scandinavian conference on image analysis, pp 325–332
30. Edward J, Murase H (1997) Appearance matching of occluded objects using coarse-to-fine adaptive masks. In: Proceedings of IEEE conference on computer vision and pattern recognition, Los Alamitos, CA, USA, pp 533–539
31. Rao RPN (1997) Dynamic appearance-based recognition. In: Proceedings of IEEE conference on computer vision and pattern recognition, San Juan, PR, USA, pp 540–546
32. Krumm J (1996) Eigenfeatures for planar pose measurement of partially occluded objects. In: Proceedings of IEEE conference on computer vision and pattern recognition, Los Alamitos, CA, USA, pp 55–60
33. Ohba K, Ikeuchi K (1997) Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects. IEEE Trans PAMI 19(9):1043–1048
34. Leonardis A, Bischof H (2000) Robust recognition using eigenimages. Comput Vis Image Understand 78(1):99–118
35. Huttenlocher DP, Lilien RH, Olson CF (1999) View-based recognition using an eigenspace approximation to the Hausdorff measure. IEEE Trans PAMI 21(9):951–955
36. Wang Z, Ben-arie J (2001) Detection and segmentation of generic shapes based on affine modeling of energy in eigenspace. IEEE Trans Image Process 10(11):1621–1629
37. Bischof H, Leonardis A (1998) Robust recognition of scaled eigenimages through a hierarchical approach. In: Proceedings of IEEE conference on computer vision and pattern recognition, Santa Barbara, CA, USA, pp 664–670
38. Schneiderman H, Kanade T (2000) A histogram-based method for detection of faces and cars. In: Proceedings of IEEE international conference on image processing, Vancouver, BC, pp 504–507
39. Mohan A, Papageorgiou C, Poggio T (2001) Example-based object detection in images by components. IEEE Trans PAMI 23(4):349–361
40. Stauffer C, Grimson E (2001) Similarity templates for detection and recognition. In: Proceedings of IEEE conference on computer vision and pattern recognition, Kauai, HI, pp I221–I228
41. Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Lett Nat 401(6755):788–791
42. Guillamet D, Vitria J (2003) Evaluation of distance metrics for recognition based on non-negative matrix factorization. Pattern Recogn Lett 24(9–10):1599–1605
43. Li SZ, Hou XW, Zhang HJ, Cheng QS (2001) Learning spatially localized, parts-based representation. In: Proceedings of IEEE conference computer vision and pattern recognition, Kauai, HI, pp I207–I212
44. Jugessur D, Dubek G (2000) Local appearance for robust object recognition. In: Proceedings of IEEE conference on computer vision and pattern recognition, Hilton Head Island, SC, USA, pp 834–839
45. Nene SA, Nayar SK (1997) A simple algorithm for nearest neighbor search in high dimensions. IEEE Trans PAMI 19(9):989–1003
46. Kakarala R, Ogunbona PO (2001) Signal analysis using a multiresolution form of the singular value decomposition. IEEE Trans Image Process 10(5):724–735
47. Uenohara M, Kanade T (1997) Use of Fourier and Karhunen–Loeve decomposition for fast pattern matching with a large set of templates. IEEE Trans PAMI 19(8):891–898
48. Ohba K, Ikeuchi K (1996) Recognition of the multi specularity objects for bin-picking task. In: Proceedings of IEEE international conference on intelligent robots and systems, Osaka, Japan, pp 1440–1447
49. Nene SA, Nayar SK, Murase H (1996) Columbia object image library (COIL-100), http://www.cs.columbia.edu/cave/research/softlib/coil-100.html. In: Technical report CUCS-006-96, Columbia University, 1996
50. Nene SA, Nayar SK, Murase H (1996) Columbia object image library (COIL-20), http://www.cs.columbia.edu/cave/research/softlib/coil-20.html. In: Technical report CUCS-005-96, Columbia University, 1996
51. Koubaroulis D, Matas J, Kittler J (2002) Evaluating colour-based object recognition algorithms using the SOIL-47 database. In: Proceedings of Asian conference on computer vision, Melbourne, Australia, pp 840–845
52. Geusebroek JM, Burghouts GJ, Smeulders AWM (2004) The Amsterdam library of object images. Int J Comput Vis 61(1):103–112
53. Chang C-Y (1999) Eigenspace methods for correlated images. PhD Dissertation, Purdue University, USA

Author Biographies

Chu-Yin Chang received the B.S. degree in mechanical engineering from National Central University, Chung-Li, Taiwan, ROC, in 1988, the M.S. degree in electrical engineering from the University of California, Davis, in 1993, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, in 1999. From 1999–2002, he was a Machine Vision Systems Engineer with Semiconductor Technologies and Instruments, Inc., Plano, TX. He is currently the Vice President of Energid Technologies, Cambridge, MA, USA. His research interests include computer vision, computer graphics, and robotics.

Anthony A. Maciejewski received the BSEE, M.S., and Ph.D. degrees from Ohio State University in 1982, 1984, and 1987. From 1988 to 2001, he was a professor of Electrical and Computer Engineering at Purdue University, West Lafayette. He is currently the Department Head of Electrical and Computer Engineering at Colorado State University. He is a Fellow of the IEEE. A complete vita is available at: http://www.engr.colostate.edu/~aam.

Venkataramanan Balakrishnan is Professor and Associate Head of Electrical and Computer Engineering at Purdue University, West Lafayette, Indiana. He received the B.Tech degree in electronics and communication and the President of India Gold Medal from the Indian Institute of Technology, Madras, in 1985. He then attended Stanford University, where he received the M.S. degree in statistics and the Ph.D. degree in electrical engineering in 1992. He joined Purdue University in 1994 after postdoctoral research at Stanford, CalTech and the University of Maryland. His primary research interests are in convex optimization and large-scale numerical algebra, applied to engineering problems.

Rodney G. Roberts received B.S. degrees in Electrical Engineering and Mathematics from Rose-Hulman Institute of Technology in 1987 and an MSEE and Ph.D. in Electrical Engineering from Purdue University in 1988 and 1992, respectively. From 1992 until 1994, he was a National Research Council Fellow at Wright Patterson Air Force Base in Dayton, Ohio. Since 1994 he has been at the Florida A&M University—Florida State University College of Engineering where he is currently a Professor of Electrical and Computer Engineering. His research interests are in the areas of robotics and image processing.

Kishor Saitwal received the Bachelor of Engineering (B.E.) degree in Instrumentation and Controls from Vishwakarma Institute of Technology, Pune, India, in 1998. He was ranked Third in the Pune University and was recipient of National Talent Search scholarship. He received the M.S. and Ph.D. degrees from the Electrical and Computer Engineering department, Colorado State University, Fort Collins, in 2001 and 2006, respectively. He is currently with Behavioral Recognition Systems, Inc. performing research in computer aided video surveillance systems. His research interests include image/video processing, computer vision, and robotics.