Antifaces: A Novel, Fast Method for Image Detection

Daniel Keren, Margarita Osadchy, and Craig Gotsman

Abstract—This paper offers a novel detection method, which works well even in the case of a complicated image collection—for instance, a frontal face under a large class of linear transformations. It is also successfully applied to detect 3D objects under different views. Call the collection of images, which should be detected, a *multitemplate*. The detection problem is solved by sequentially applying very simple filters (or *detectors*), which are designed to yield *small* results on the multitemplate (hence, "antifaces"), and *large* results on "random" natural images. This is achieved by making use of a simple probabilistic assumption on the distribution of natural images, which is borne out well in practice. Only images which passed the threshold test imposed by the first detector are examined by the second detector, etc. The detectors are designed to act independently so that their false alarms are uncorrelated; this results in a false alarm rate which decreases exponentially in the number of detectors. This, in turn, leads to a very fast detection algorithm. Typically, $(1+\delta)N$ operations are required to classify an *N*-pixel image, where $\delta < 0.5$. Also, the algorithm requires no training loop. The algorithm's performance compares favorably to the well-known eigenface and support vector machine based algorithms, but is substantially faster.

Index Terms—Image detection, smoothness, distribution of natural images, rejectors.

1 Introduction

In computer vision, the well-known template detection problem can be formalized as: Given an image T (the *template*) and a (usually much larger) image P, determine whether there are instances of T in P, and if so, where. A typical scenario is: Given a photograph of a face and a large image, determine if the face appears in the image.

This problem may be solved by various methods such as cross-correlation or Fourier-based techniques [25], [3], [19]. A more challenging problem is what we call *multitemplate detection*. Here, we are given not one template T, but a *class* of templates $\mathcal{T}$ (which we call a *multitemplate*), and are required to answer the more general question: Given a large image P, locate all instances of *any* member of $\mathcal{T}$ within P. Obviously, if $\mathcal{T}$ can be well-represented by n templates, we could apply the standard template detection techniques n times and take the union of the results. This naive approach, however, breaks down in complexity for large n. The goal of this research is to develop an efficient algorithm for multitemplate detection.

Typical cases of interest are:

- Given an image, locate all instances of human faces in it.
- Given an aerial photograph of an airfield, locate all instances of an airplane of a given type in it. If we do not know the angle at which the airplanes are parked and the position from which the photograph was taken, then we have to locate not a fixed image of the airplane, but some affinely distorted version of it. If the photograph was taken from a relatively low altitude, we may have to look for perspective distortions as well. In this case, the multitemplate consists of a collection of affinely (perspectively) distorted versions of the airplane, and it can be well-approximated by a finite collection of distorted versions, sampled closely enough in transformation space (obviously, one will have to limit the range of distortions; say, allow scale changes only at a certain range, etc.).
- Locate different views of a three-dimensional object in a given image.

1.1 Structure of the Paper

After surveying some of the related research, we proceed to define some relevant concepts and outline the idea behind the suggested detection scheme. Then, the mathematical foundations for the antiface algorithm are laid. Following that, some experimental results are presented, and compared with eigenface, Fisher linear discriminant, and support vector machines-based methods.

1.2 Previous Work

Most detection algorithms may be classified as either intensity-based or feature-based. Intensity-based methods operate directly on the pixel gray-level intensities. In contrast, feature-based methods first extract various geometric cues from the raw image, then perform higher-level reasoning on this geometric information.

Previous work on multitemplate detection includes an extension of Fourier-based techniques [18]. There is also a large body of work on the recognition of objects distorted under some geometric transformation group, using

- *D. Keren and M. Osadchy are with the Department of Computer Science, University of Haifa, Haifa 31905, Israel. E-mail: {dkeren, gamer}@cs.haifa.ac.il.*
- *C. Gotsman is with the Department of Computer Science, Technion, Technion City, Haifa 32000, Israel. E-mail: gotsman@cs.technion.ac.il.*

Manuscript received 21 June 2000; revised 5 Jan. 2001; accepted 19 Jan. 2001. Recommended for acceptance by D. Kriegman.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 112312.

0162-8828/01/$10.00 © 2001 IEEE

invariants [34]. Some intensity-based methods use moment invariants for recognition of objects under Euclidean or affine transformations [11]. One difficulty with these methods is that one has to compute the local moments of many areas in the input image. Also, moment-based methods cannot handle more complex transformations (e.g., there are no moment invariants for projective transformations or among different views of the same three-dimensional object).

Feature-based algorithms [10] have to contend with the considerable difficulty of locating features in the image. Methods that use differential invariants [34] and, thus, require computing derivatives, have to overcome the numerical difficulties involved in reliably computing such derivatives in noisy images.

Of the intensity-based methods for solving the multitemplate detection problem, the *eigenface* method [24], [9], [29], [30] has drawn a great deal of attention. This method approximates the multitemplate $\mathcal{T}$ by a low-dimensional linear subspace F, usually called the *face space*. Images are initially classified as potential members of $\mathcal{T}$, if their distance from F is smaller than a certain threshold. The images which pass this test are projected on F and these projections are compared to those in the training set.

The eigenface method can be viewed as an attempt to model $\mathcal{T}$'s distribution. Other work on modeling this distribution includes the study of the within-class versus "general" scatter [2], [27], [26] and a more elaborate modeling of the probability distribution in the face class [12]. In [13], eigenfaces were combined with a novel search technique to detect 3D objects and, also, recover their pose and the ambient illumination; however, it was assumed that the objects (from the COIL database) were already segmented from the background and recognition was restricted to that database.

The eigenface method has been rather successful for various detection problems such as detecting frontal human faces. However, our experiments suggest that once a large class of transformations comes into play—for instance, if one tries to detect objects under arbitrary rotation and possibly other distortions—the eigenface method runs into problems. This was confirmed by one of the first researchers to apply eigenfaces to detection [31].

In an attempt to apply the eigenface principle to detection under linear transformations [32], a version of the method was applied to detect an object with strong high-frequency components in a cluttered scene. However, the range of transformations was limited to rotation and only at the angles -50^o to 50^o. The dimension of the face space used was 20. We will show results for a far more complicated family of transformations, using a faster algorithm.

Neural nets have been applied, with considerable success, to the problem of frontal face detection [21] and, also, of faces under unknown rotation [22]. It is not clear whether the methods used in [22] can be extended to more general transformation groups than the rotation group, as the neural net constructed there is trained to return the rotation angle; for a family of transformations with more than one degree of freedom, both the training and the detection become far more complicated because the size of the training set and the net's set of responses, grow exponentially with the number of degrees of freedom.

Support vector machines (SVM's) [16], [14], [15], [20], [23] were introduced by Vapnik [33] and can be viewed as a mechanism to find an optimal separating hyperplane, either in the space of the original variables, or in a higher-dimensional "feature space." The feature space consists of various functions of the components of the original $\mathbf{t}$ vectors, such as polynomials in these components, and allows for more powerful detection. The optimal hyperplane maximizes the margin between training sets for the multitemplate $\mathcal{T}$ and its complement.

An SVM consists of a function G which is applied to each candidate image $\mathbf{t}$ and it classifies it as a member of $\mathcal{T}$ or not, depending on the value of $G(\mathbf{t})$. A great deal of effort has been put into finding such a function which optimally characterizes $\mathcal{T}$. A typical choice is

$$G(\mathbf{t}) = sgn\left(\sum_{i=1}^{l} \lambda_i y_i K(\mathbf{t}, \mathrm{x_i}) + b\right),$$

where $\mathbf{t}$ is the image to be classified, $\mathbf{x_i}$ are the training images, y_i is 1 or -1 depending on whether $\mathbf{x_i}$ is in $\mathcal{T}$ or not, and $K()$ a "kernel function" (for example, $K(\mathbf{t}, \mathbf{x_i}) = \exp(-\|\mathbf{t} - \mathbf{x_i}\|^2)$). Usually, only a relatively small number of the $\mathbf{x_i}$ are used and these $\mathbf{x_i}$ are called the *support vectors*. Thus, the speed of SVM's depends to a considerable extent on the number of support vectors. The λ_i are typically recovered by solving a quadratic programming problem.

As opposed to SVM's and neural nets, the method suggested here does not require a training loop on negative examples because it makes an assumption on their statistics—which is borne out in practice—and uses it to reduce false alarms (false alarms are cases in which a nonmember of $\mathcal{T}$ is erroneously classified as a member).

1.3 A Short Description of the Motivation Behind the Antiface Algorithm

A basic notion in detection and pattern recognition (see [5] for a general introduction and, also, [1]) is that of a *discriminant function*. For instance, if one wishes to quickly determine whether a point in the plane, (x, y), belongs to the unit circle, the most efficient way is to compute the value of $x^2 + y^2 - 1$. That is, we make use of the fact that there exists a simple function, which assumes a value of zero on the set to be detected—and *only* on it. We could also use this fact to test whether a point is close to the unit circle.

Generalizing, we may look at discriminant functions as describing a set by

$$A = \bigcap_{i=1}^{m} f_i^{-1}[-\epsilon_i, \epsilon_i], \tag{1}$$

where f_i are the discriminating functions and the ϵ_i are small. For the unit circle, for instance, one function is required: $f_1 = x^2 + y^2 - 1$.

Thus, to test whether a point (or, in our case, an image viewed as a point in high-dimensional Euclidean space) $\mathbf{x}$, belongs to a multitemplate $\mathcal{T}$, one has to verify that, for every $1 \leq i \leq m$, $|f_i(\mathbf{x})| \leq \epsilon_i$. The decision process can be shortened by first checking the condition for f_1 and applying f_2 only to the images for which $|f_1(\mathbf{x})| \leq \epsilon_1$, etc. It will be shown later in the paper (Section 3), that this progressive detection very substantially reduces the running time.

This very general scheme offers an attractive algorithm for detecting $\mathcal{T}$, if the following conditions hold:

- $A \supseteq \mathcal{T}$. This is crucial, as $\mathcal{T}$ must be detected.
- m is small.
- The discriminating functions f_i are easy to compute.
- If $\mathbf{y} \notin \mathcal{T}$, there is a small probability that $|f_i(\mathbf{y})| \leq \epsilon_i$ for every i.

Since this work addresses image detection, from here on, the term *detector* will replace "discriminating function." See [1], in which the notion of a "rejector" is defined; it differs from the work presented here mainly in the modeling of the "nonobjects."

Images are large; it is therefore preferable to use simple detectors. Let us consider then detectors which are linear and act as inner products with a given image (viewed as a vector). For this to make sense, the detectors have to be normalized, so assume that they are of unit length and zero average. If $| < \mathbf{d}, \mathbf{t} > |$ is very small for every $\mathbf{t} \in \mathcal{T}$, then $f(\mathbf{y}) = | < \mathbf{d}, \mathbf{y} > |$ is a candidate detector for $\mathcal{T}$. However, if we choose such a few "random" $\mathbf{d}_\mathbf{i}$s, this naive approach will fail, as $| < \mathbf{d}_\mathbf{i}, \mathbf{y} > |$ is also very small for many images y which are not close to any member of $\mathcal{T}$.

Let us demonstrate this by an example. The object that has to be detected is a pocket calculator, photographed at an unknown pose, from an unknown angle, and from a range of distances which induces a possible scaling factor of about $0.7 - 1.3$ independently at both axes. Thus, $\mathcal{T}$ consists of many affinely distorted images of the pocket calculator. Naively, we may try to use, as detectors, a few unit vectors, whose inner product with every member of $\mathcal{T}$ is small; they are easy to find, applying a standard SVD decomposition of $\mathcal{T}$'s scatter matrix, and using the eigenvectors with the smallest eigenvalues. In Fig. 1, we show the result of this naive algorithm which—not surprisingly—fails.

Fig. 1 demonstrates that it is not enough for the detectors to yield small values on the multitemplate $\mathcal{T}$: while this is satisfied by the detectors depicted in Fig. 1, the detection results are very bad. Not only are many false alarms present, but the correct location is missed, due to noise and the instability of the detectors. More specifically, the detection fails because the detectors *also* yield very small results on many subimages which are not members of $\mathcal{T}$ (nor close to any of its members). Thus, the detectors have to be modified so that they will not only yield small results on $\mathcal{T}$'s images, but large results on "random" natural images.

To the rescue comes the following probabilistic observation: Most natural images are *smooth*. As will be formally proven in the sequel, **the absolute value of the inner product of two smooth vectors is, on the average, large**. If $\mathbf{d}$ is a candidate for a detector to the multitemplate $\mathcal{T}$, suppose that not only is $| < \mathbf{d}, \mathbf{t} > |$ small for $\mathbf{t} \in \mathcal{T}$, but also that $\mathbf{d}$ is smooth. Then, if $\mathbf{y} \notin \mathcal{T}$, there is a high probability that $| < \mathbf{d}, \mathbf{y} > |$ will be large; this allows us to reject $\mathbf{y}$, that is, determine that it is not a member of $\mathcal{T}$.

In the spirit of the prevailing terminology, we call such detectors $\mathbf{d}$ "antifaces" (this does not mean that detection is restricted to human faces). Thus, a candidate image $\mathbf{y}$ will be rejected if, for some antiface $\mathbf{d}$, $| < \mathbf{d}, \mathbf{y} > |$ is larger than some $\mathbf{d}$-specific threshold. This is a very simple process which can be quickly implemented by a rather small number of inner products. Since the candidate image has to satisfy the conditions imposed by *all* the detectors, it is enough to apply the second detector only to images which passed the first detector test, etc. In all cases tested, this resulted in a number of operations less than $1.5N$ operations, for an N-pixel candidate image. In the typical case in which all the subimages of a large image have to be tested, the first detector can be applied by convolution.

2 The "Antiface" Method: Mathematical Foundation

To recap, for a multitemplate $\mathcal{T}$, the "antiface detectors" are defined as vectors satisfying the following three conditions:

- The absolute values of their inner product with $\mathcal{T}$'s images are small.
- They are smooth, which results in the absolute values of their inner product with "random images" being large; this is the characteristic which enables the detectors to separate $\mathcal{T}$'s images from random images. This will be formalized in Section 2.1.
- They act in an independent manner which implies that their false alarms are uncorrelated. As we shall prove, this does not mean that the inner product of different detectors is zero, but implies a slightly different condition. The independence of the detectors is crucial to the algorithm's success, as it results in a number of false alarms which decreases exponentially in the number of detectors. This is explained in Section 2.2.

Once the detectors are found, the detection process is very easy to implement: An image is classified as a member of $\mathcal{T}$ iff the absolute value of its inner product with each detector is smaller than some (detector specific) threshold. Typically, the threshold was chosen as twice the maximum over the absolute values of the inner products of the given detector with the members of a training set for $\mathcal{T}$. This factor of two allows detection not only of the members of the training set, but also of images which are close to them.

A schematic description of the geometry behind antifaces is presented in Fig. 2. The algorithm's "positive set" (the images it classifies as members of the multitemplate), is orthogonal to the direction around which random images cluster, hence, there are relatively few false alarms. In the

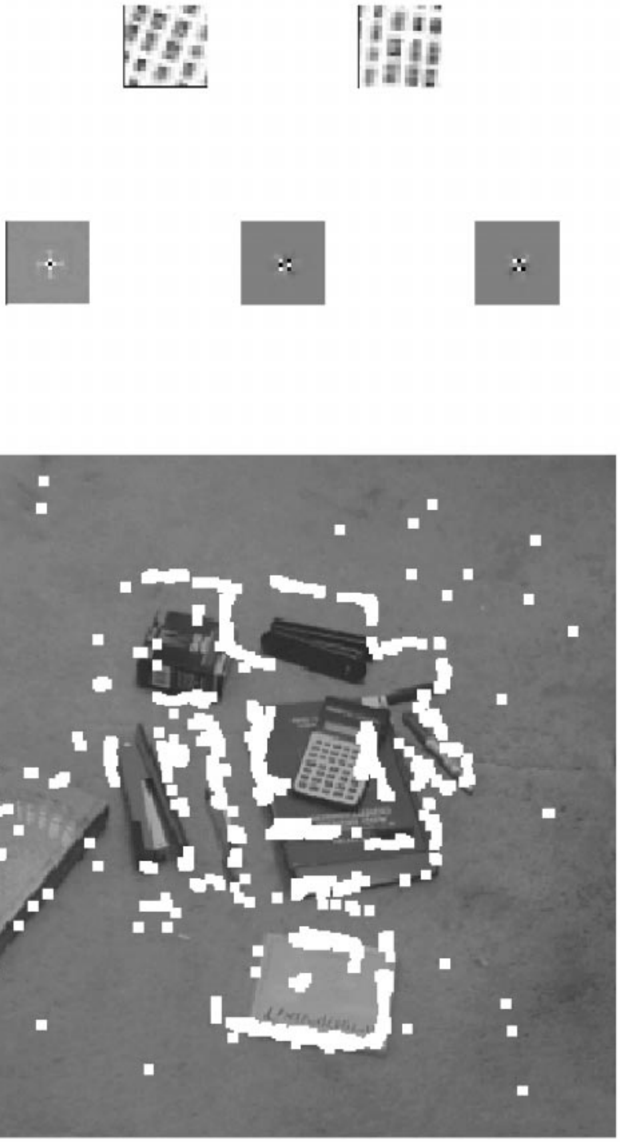

Fig. 1. Top: Two of the members of the calculator multitemplate which consists of affinely distorted versions of the key area in a pocket calculator. Middle: Some of the "naive" detectors for the pocket calculator multitemplate. Note that they contain strong high-frequency components. Bottom: Failure of "naive" detectors to find the target. Detection is marked by a small bright square at the upper left corner of the detected image region; the image has been artificially darkened in order to make the detection results more visible.

eigenface method, however, many random images will pass the initial test which accepts an image based on its distance from the face space; this is because the leading components in the principal component decomposition of the multitemplate, will usually be closely aligned with the leading components in the decomposition of random images—as both are largely smooth. While these images may be filtered out during the later stages of the eigenface algorithm, they still incur a heavy computational price.

2.1 Computing the Expectation of the Inner Product

We now proceed to prove that the absolute value of the inner product of two "random" natural images is large (for the statement to make sense, assume that both images are of zero mean and unit norm). The Boltzman distribution, which has proven to be a reasonable model for natural images [8], [6], assigns to an image $\mathbf{I}$ a probability proportional to the exponent of the negative of some "smoothness measure" for $\mathbf{I}$. Usually, an expression such as $\iint(\mathbf{I}_x^2+\mathbf{I}_y^2)dxdy$, or $\iint(\mathbf{I^2}_{xx}+2\mathbf{I}_{xy}^2+\mathbf{I}_{yy}^2)dxdy$, is used [8], [28]. It is preferable, for the following analysis, to work in the frequency domain since then the smoothness measure operator is diagonal, hence more manageable. The smoothness of an $n \times n$ image $\mathbf{I}$, denoted $S(\mathbf{I})$, is defined by

$$S(\mathbf{I}) = \sum_{(k,l)\neq(0,0)}^{n} (k^2+l^2)\mathcal{I}^2(k,l) \tag{2}$$

and its probability is defined, following the Boltzman distribution, as

$$Pr(\mathbf{I}) \propto \exp(-S(\mathbf{I})), \tag{3}$$

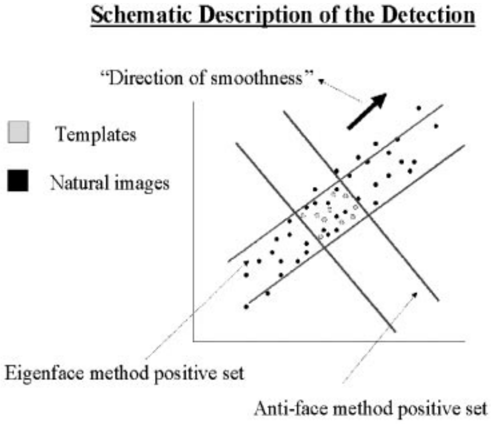
Schematic Description of the Detection
"Direction of smoothness"
Templates
Natural images
Eigenface method positive set
Anti-face method positive set

Fig. 2. Schematic description of the antiface algorithm. Random natural images cluster around the "direction of smoothness"—that is, with high probability, they lie in a double cone extending between the origin and the point $(1, 1, 1 \ldots 1)$ (the smoothest image) on one side, and between the origin and $(-1, -1, -1 \ldots -1)$ on the other side. To separate the multitemplate from random images, the detector's "positive set" should therefore be as orthogonal as possible to the "direction of smoothness." This is achieved by using a smooth detector which is closely aligned with the null space of the multitemplate and defining its "positive set" as the set of images whose inner product with the detector is smaller in absolute value than a given threshold. The eigenfaces method will incur many false alarms, as the so-called "face space" will contain many smooth random images since its major axes are closely aligned with the "direction of smoothness."

where $\mathcal{I}(k, l)$ are the DCT (Discrete Cosine Transform) coefficients of $\mathbf{I}$. Since the images are normalized to zero mean, $\mathcal{I}(0, 0) = 0$. This definition is clearly in the spirit of the continuous, integral-based definitions, and assigns higher probabilities to smoother images. Hereafter, when referring to "random images," we shall mean images randomly sampled from this probability space. Now, it is possible to formalize the observation "the absolute value of the inner product of two random images is large." For a given image $\mathbf{F}$ of size $n \times n$, the expectation of the square of its inner product with a random image equals

$$E[< \mathbf{F}, \mathbf{I} >^2] = \int\limits_{\mathcal{R}^{n\times n}} < \mathbf{F}, \mathbf{I} >^2 Pr(\mathbf{I}) d\mathbf{I}$$

using Parseval's identity, this can be computed in the DCT domain. Substituting the expression for the probability (3) and denoting the DCT transforms of $\mathbf{F}$ and $\mathbf{I}$ by $\mathcal{F}$ and $\mathcal{I}$, respectively, we obtain

$$\int\limits_{\mathcal{R}^{n\times n-1}} \left(\sum_{(k,l)\neq(0,0)} \mathcal{F}(k,l)\mathcal{I}(k,l) \right)^2 \exp\left(- \sum_{(k,l)\neq(0,0)}^{n} (k^2 + l^2)\mathcal{I}^2(k,l) \right) d\mathcal{I},$$

which, after some manipulations (see the Appendix), turns out to be proportional to

$$\sum_{(k,l)\neq(0,0)} \frac{\mathcal{F}^2(k,l)}{(k^2 + l^2)^{3/2}} \tag{4}$$

since the images are normalized to unit length, it is obvious that, for the expression in (4) to be large, the dominant values of the DCT transform $\{\mathcal{F}(k, l)\}$ should be concentrated in the small values of k, l—in other words, that $\mathbf{F}$ be smooth.

This theoretical result is well-supported empirically. In Fig. 3, the empirical expectation of $< \mathbf{F}, \mathbf{I} >^\mathbf{2}$ is plotted against (4). The expectation was computed for 5,000 different $\mathbf{F}$, by averaging their squared inner products with 15,000 subimages of natural images. The size was 20×20 pixels. The figure demonstrates a reasonable linear fit between (4) and the empirical expectation.

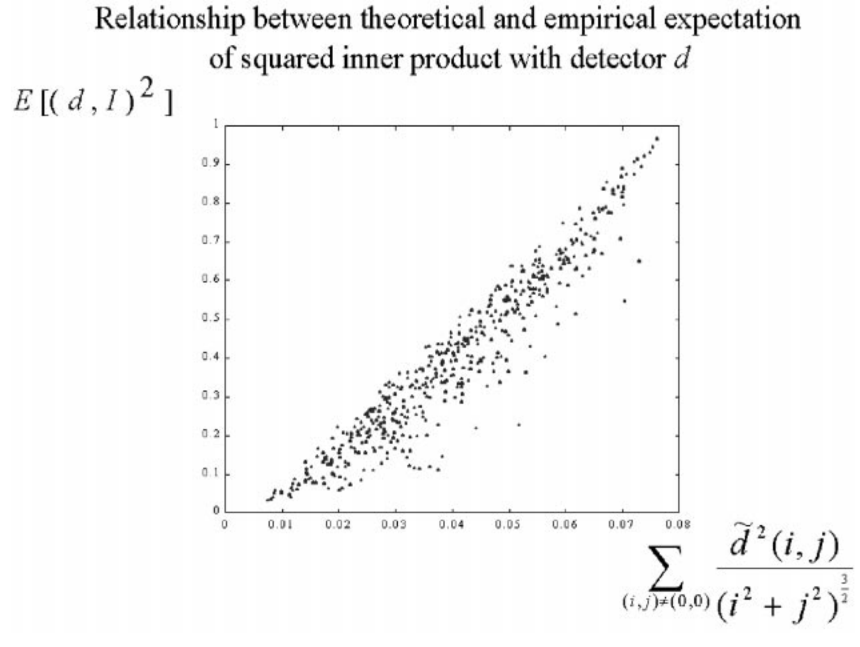

Fig. 3. Empirical verification of (4). $\tilde{d}$ denotes the DCT transform of d.

2.2 Constructing Independent Detectors

It is unreasonable to expect that one detector can detect $\mathcal{T}$ without many false alarms. This is because, for a single detector $\mathbf{d}$, although $| < \mathbf{d}, \mathbf{y} > |$ is large on the average for a random image $\mathbf{y}$, there will always be many random images $\mathbf{I}$ such that $| < \mathbf{d}, \mathbf{I} > |$ is small, and these images will be erroneously classified as members of $\mathcal{T}$. The optimal remedy for this is to apply a few detectors which act *independently*; this implies that if the false alarm rate (defined as the percentage of false alarms) of $\mathbf{d_1}$ is p_1, and that of $\mathbf{d_2}$ is p_2, then the false alarm rate for both detectors will be $p_1 p_2$. Since the entire detection scheme rests on the probability distribution defined in (3), the notion of independence is equivalent to the requirement that the two random variables, defined by $\mathbf{I} \rightarrow < \mathbf{I}, \mathbf{d_1} >$ and $I \rightarrow < \mathbf{I}, \mathbf{d_2} >$, be independent, or

$$\int_{\mathcal{R}^{n \times n - 1}} < \mathbf{I}, \mathbf{d_1} >< \mathbf{I}, \mathbf{d_2} > Pr(\mathbf{I}) d\mathbf{I} = 0$$

denote this integral by $< \mathbf{d_1}, \mathbf{d_2} >^*$; it turns out (see the Appendix) to be

$$< \mathbf{d_1}, \mathbf{d_2} >^* = \sum_{(k,l) \neq (0,0)} \frac{\mathcal{D}_1(k,l)\mathcal{D}_2(k,l))}{(k^2 + l^2)^{3/2}}, \qquad (5)$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$ are the DCT transforms of $\mathbf{d_1}$ and $\mathbf{d_2}$.

2.3 Computing the Detectors

To find the first antiface detector, $\mathbf{d_1}$, the following optimization problem should be solved (here, we assume that $\mathcal{T}$ is the given training set for the multitemplate):

1. $\mathbf{d_1}$ has to be of unit norm.
2. $| < \mathbf{d_1}, \mathbf{t} > |$ should be small, for every image $\mathbf{t}$ in $\mathcal{T}$. Note that every input image is also normalized, for the condition to make sense.
3. $\mathbf{d_1}$ should be as smooth as possible under the first and second constraints, which will ensure that the expression in (4) will be large. We have also tried maximizing the expression in (4) directly, but that did not result in any performance improvement. As a matter of fact, in some cases, it turned out that opting for a smoother detector is slightly better than directly maximizing (4)—probably because a smoother detector acts more continuously; since the detector is built using a training set, it is desirable that it act continuously, and, thus, yield small results also on images which are close to the training set.

The solution we implemented proceeds as follows: First, choose an appropriate value for

$$\max_{\mathbf{t} \in \mathcal{T}} | < \mathbf{d_1}, \mathbf{t} > |;$$

experience has taught us that it doesn't matter much which value is used, as long as it is substantially smaller than the absolute value of the inner product of two random images. Usually, for images of size 20×20, we have chosen this maximum value—denoted by M—as 10^{-5}. If it is not possible to attain this value—which will happen if $\mathcal{T}$ is too complicated—choose a larger M. Next, minimize

$$\max_{\mathbf{t} \in \mathcal{T}} | < \mathbf{d_1}, \mathbf{t} > | + \lambda S(\mathbf{d_1})$$

and, using a binary search on λ, set it so that

$$\max_{\mathbf{t} \in \mathcal{T}} | < \mathbf{d_1}, \mathbf{t} > | = M.$$

We have used the Nelder-Mead method [17] for the optimization. The optimization is performed in the DCT domain, and the inverse DCT transform of the optimum is the desired detector (note that the detection itself is carried out directly on the images; the DCT domain is used only in the offline computation of the detectors).

After $\mathbf{d_1}$ is found, it is straightforward to recover $\mathbf{d_2}$; the only difference is the additional condition $< \mathbf{d_1}, \mathbf{d_2} >^* = 0$ (5) and it is easy to incorporate this condition into the optimization scheme. The other detectors are found in a similar manner.

Note that $\mathbf{d_1}$ has to satisfy fewer constraints than the other detectors, hence, it is smoother than them. Therefore, it is applied as the first detector, as it will filter out more input images than the other detectors. In the same vein, $\mathbf{d_2}$ is smoother than $\mathbf{d_3}$, hence, it is applied to the images which passed the threshold imposed by $\mathbf{d_1}$, etc.

2.3.1 A Faster Algorithm for Suboptimal Detectors

Although the detectors are computed offline, it may be desirable, in some cases, to use a faster algorithm. Then, one may replace the target function

$$\max_{\mathbf{t} \in \mathcal{T}} | < \mathbf{d_1}, \mathbf{t} > | + \lambda S(\mathbf{d_1})$$

with the simpler

$$\sum_{\mathbf{t} \in \mathcal{T}} < \mathbf{d_1}, \mathbf{t} >^2 + \lambda S(\mathbf{d_1})$$

which may be optimized as above (yielding a different λ). While not optimal, the target function is now quadratic and the detectors can be found by a standard SVD routine.

Empirical results indicate that, typically, if n optimal antiface detectors achieve a certain detection rate, then about $1.3n$ suboptimal detectors are required for achieving the same rate. The results in Section 3 were all obtained using suboptimal detectors.

3 Experimental Results

The antiface method was tested both on synthetic and real inputs. In Section 3.1, it is compared to the eigenface method regarding the problem of detecting a frontal face subject to increasingly complex families of transformations. In these experiments, the test images were synthetically created. In the other experiments, the method was applied to detect various objects in real images: A pocket calculator which is nearly planar and the well-known COIL database of 3D objects, photographed at various poses.

Last, we briefly address the problem of detection under varying illumination. The antiface method offers an attractive solution, which proceeds by including the effects

TABLE 1
Performance of the Eigenface and Antiface Algorithms as a Function of the Multitemplate's Complexity

Algorithm's Performance	Rotation	Rotation + Scale	Linear
Number of Eigenvalues Required for 90% Energy	13	38	68
Eigenface Performance: Dimension of Face Space Required for Accurate Detection	12	74	145
Anti-Face Performance: Number of Detectors Required for Accurate Detection	3	4	4

of different illumination conditions in the multitemplate; this automatically cancels the illumination effect, allowing fast, illumination invariant detection.

The number of detectors required for each experiment is provided. Note that, since every detector acts only on images which passed the thresholds imposed by the previous detectors, the average running time for an N-pixel input image is much smaller than kN, where k is the number of detectors.

3.1 Performance as a Function of Multitemplate's Complexity

In order to test the performance of the antiface method with multitemplates of increasing complexity, the following three multitemplates have been created, each of which consists of a family of transformations applied to the frontal image of the "Esti" face (20×20 pixels):

- Rotation only.
- Rotation and uniform scale at the range 0.7 to 1.3.
- The set of linear transformations spanned by rotations and independent scaling at the x and y axis, at the range 0.8 to 1.2.

In order to estimate the complexity of these multitemplates, the scatter matrix for a training set of each was built and the number of largest eigenvalues whose sum of squares equals 90 percent of the sum of squares of all 400 eigenvalues, was computed. This is a rough measure of the "linear complexity" of the multitemplate.

Ten images from each multitemplate were then superimposed on an image consisting of 400 human faces, each 20×20 pixels and both the eigenface and antiface algorithms were applied. These 10 images were not in the training set.

Interestingly, while the eigenface method's performance decreased rapidly as the multitemplate's complexity increased, there was hardly a decrease in the performance of the antiface method. Table 1 summarizes the results; by "accurate detection," we mean that all the "Esti" faces (and them alone) were correctly detected.

The high linear dimension of the multitemplates is well in accordance with the observation in [4], concerning the complexity of an image set containing affinely distorted human faces.

3.1.1 Independence of the Detectors

For the case of linear transformations (the most complicated multitemplate), the false alarm rates for the first, second, and third detectors, were $p_1 = 0.0518$, $p_2 = 0.0568$, and $p_3 = 0.0572$, respectively; the false alarm rate for the three combined was 0.00017—which is almost equal to $p_1 p_2 p_3$. This indicates that the detectors indeed act independently. With four detectors, there were no false alarms.

3.1.2 Detectors and Results

Some of the images in the "Esti" multitemplate are shown, as well as the first six detectors (see Fig. 4), the detection result of the antiface method (Figs. 5, 6, 7, and 8), and the result of the eigenface method with a face space of dimension 100 (Fig. 9). The first six detectors for the calculator multitemplate are also depicted in Fig. 4 (compare to Fig. 1).

3.2 Detection of Pocket Calculator

In this set of experiments, the problem of detecting a pocket calculator at an unknown pose, photographed from different angles and distances, was tackled (see Fig. 1). Here, too, the antiface method performed well and eight detectors sufficed to recover the calculator in all the experiments, without false alarms, which was substantially better than the eigenface method. The average number of detectors per pixel was 1.45 (Fig. 10).

3.3 Comparison with Fisher Linear Discriminant

The Fisher Linear Discriminant (FLD) is a well-known tool for supervised clustering [5]. Given training data for two classes C_1 and C_2, a vector $\mathbf{v}$ is sought so that when C_1 and C_2 are projected on the subspace spanned by $\mathbf{v}$, the ratio between the distance of the centers of the projections and their scatter is maximal.

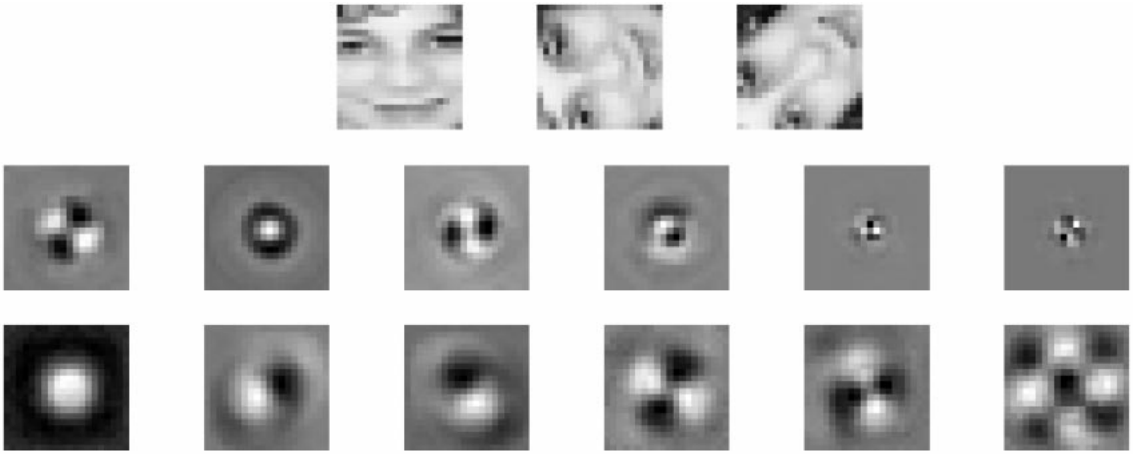

Fig. 4. Top: Sample 20×20 pixel templates, "Esti" face under various linear transformations. The range of rotations was $0 - 2\pi$, sampled at $\pi/90$ intervals, and at each angle the image was independently scaled in the x and y axes, at a range of $0.8 - 1.2$, sampled at 0.05 intervals. Altogether, there were $180 \times 9 \times 9 = 14,580$ images in the training set. Middle: The first six antiface detectors for the "Esti" multitemplate. Bottom: The first six antiface detectors for the calculator multitemplate.

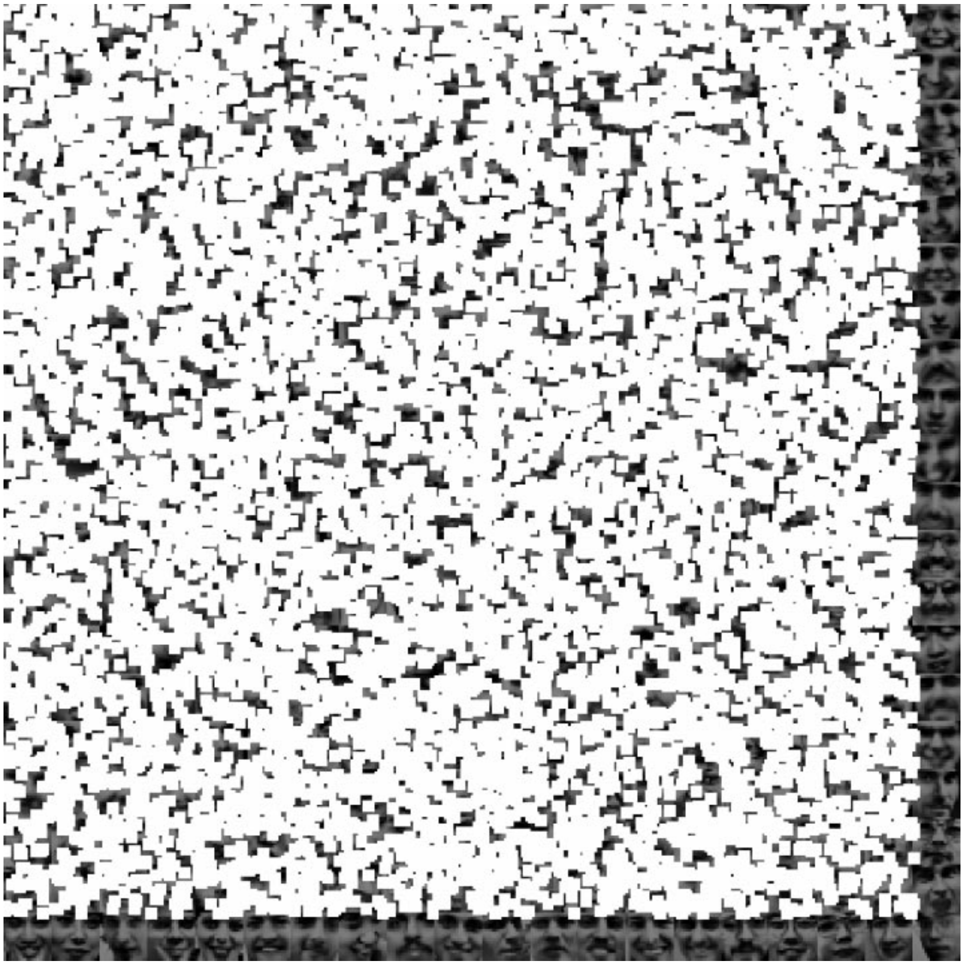

Fig. 5. Detection of "Esti" face, antiface method, one detector; note many false alarms (detection is marked by a small white square at the upper left corner of the detected subimage; there are no false alarms at the bottom and right stripes of the image, as the detection is only applied to the 20×20 subimages which are entirely inside the compound image). The multitemplate consisted of 20×20 images of a face ("Esti"), subject to the aforementioned class of linear transformations. There was no assumption on the location of the sought faces (hence, false alarms, which consist of portions of various faces, were also possible, as all subimages were tested). The transformations used to create the 10 "Esti" faces in the compound image were not part of the training set used to construct the detectors. Other face images courtesy of Henry Rowley.

In order to compare FLD to the antiface method, we have used as training sets the "Esti" multitemplate (Section 3.1), and a large number of random images. In Fig. 11, the distributions of the classes after projection on the optimal

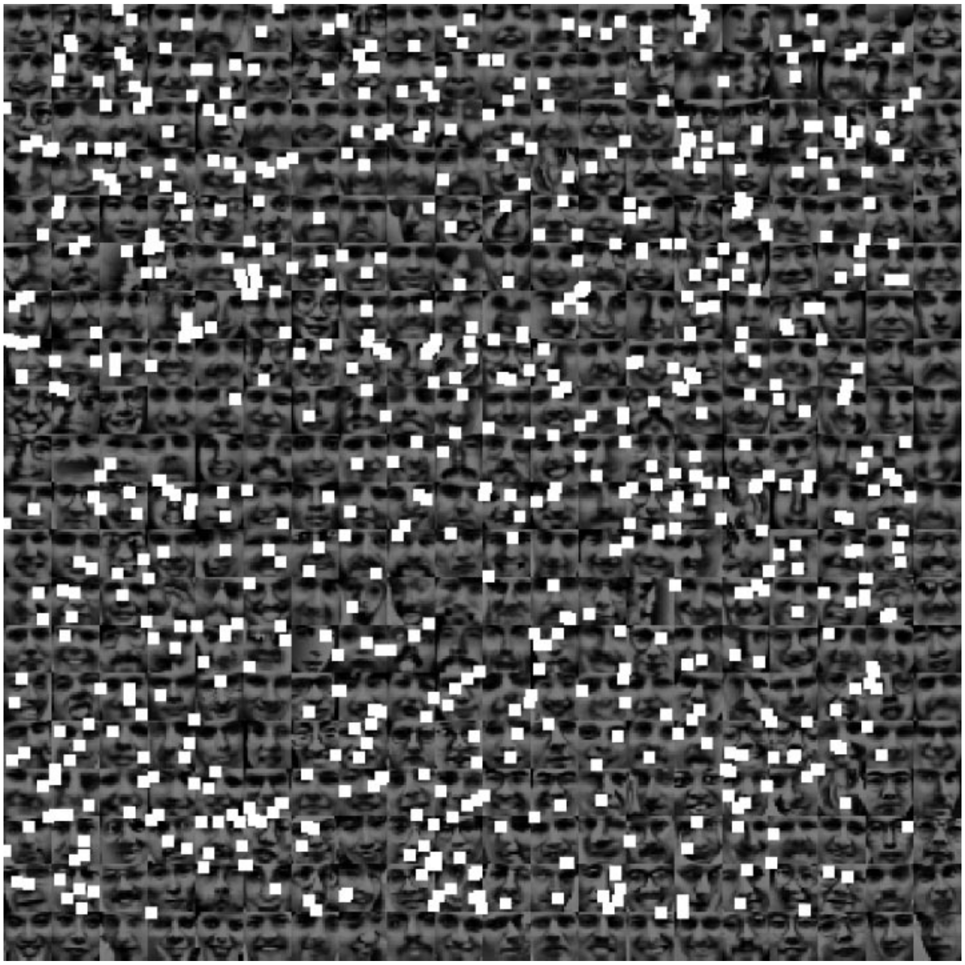

Fig. 6. Detection of "Esti" face, antiface method, two detectors; note the sharp decrease in the number of false alarms, relative to one detector. The second detector is applied only to subimages which passed the first detector's test.

v-subspace are shown. The "Esti" projections are depicted by asterisks and the training set of random images by a plain line (the middle of the three narrower Gaussian-like distributions). It is clear that, if we allow no false negatives (that is, each instance of the "Esti" multitemplate must be correctly classified), then it is impossible to choose a threshold which will enable reasonable classification since FLD will recognize practically every input image as an "Esti" instance.

It is interesting to note that FLD does not fail in the learning stage. To test its capacity to learn the concept of a random image, two additional sets of random images (that is, smooth, or "random in the Boltzman sense") were projected on the same vector found using the first set (Fig. 11). The projections of the three sets of random images have a similar distribution. Thus, FLD fails not because it cannot learn the concept of a random image from a training set, but because there is no satisfactory way to separate the multitemplate from random images by such a projection.

3.4 Detection of Objects from the COIL Database

The antiface algorithm was also applied to images from the well-known COIL database which consists of 100 three-dimensional objects, placed on a rotating table, and photographed from 72 different directions, at 5^o intervals (see www.cs.columbia.edu/CAVE/research/softlib/coil-100.html). The problem of detecting the COIL objects was addressed, for instance, in [13], [16], [20]. In [16], [20], the authors built an SVM classifier for each image pair and ran a tournament-like scheme for detecting the correct object. In [16], a relatively large amount of noise and distortion was added to the images, and the system still performed well.

In [16], 36 views of each object were used for training and the number of support vectors is specified as 1/3 to 2/3 of the training images. The same approach was undertaken in [20], which also includes extensive testing of SVM's with training sets of varying size. Roobaert and Van Hulle [20] also compare SVM's to a nearest-neighbor classifier and to the system developed by Murase and Nayar at Columbia University [13]; for 30 objects and 36 or eight views per object, SVM's and Murase and Nayar's method's performance is quite similar, and both do better than the nearest-neighbor classifier.

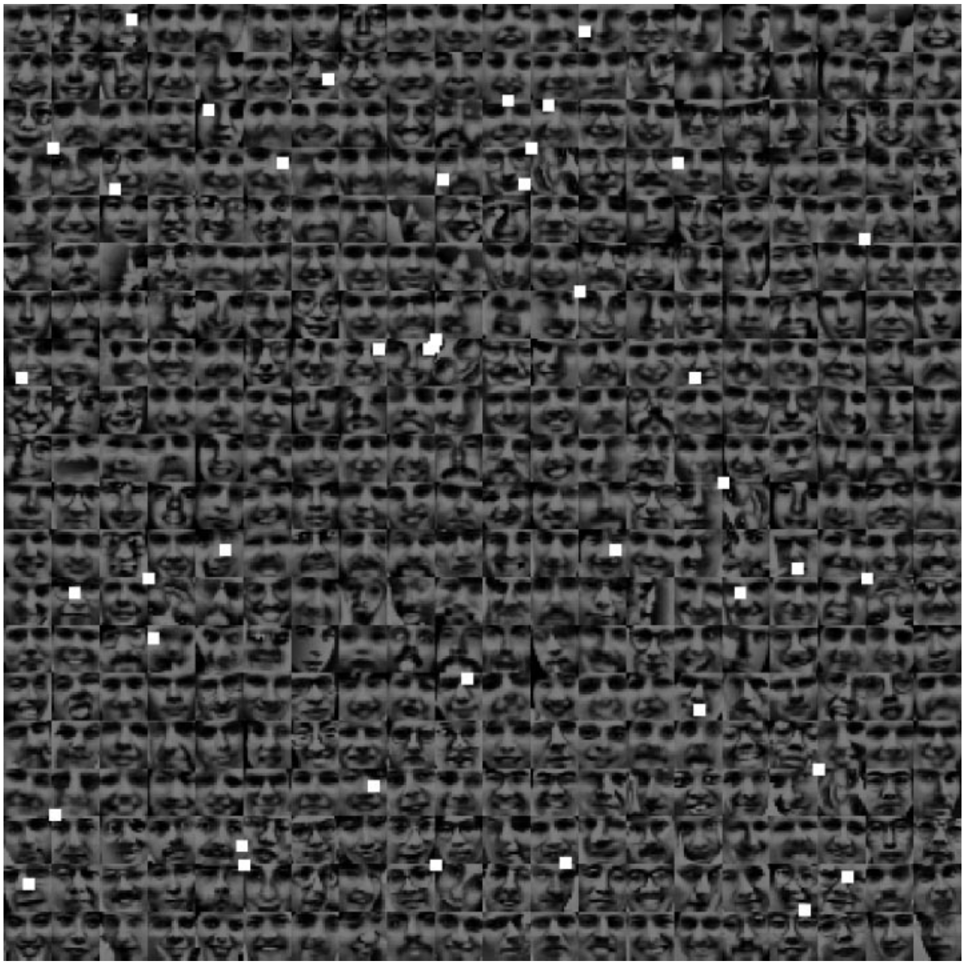

Fig. 7. Detection of "Esti" face, antiface method, three detectors; note the sharp decrease in the number of false alarms, relative to two detectors. The third detector is applied only to subimages which passed the first and second detector's test.

In [13], [16], [20], it was assumed that the objects were already segmented from the background and that the only possible input images were the ones in the COIL database. The experiment presented here is more general, in that it attempts to detect each object without any assumption on the background. We have built, for every object, a detector which is trained on the multitemplate of the 36 images at angles which are multiples of 10^o (as in [16]) and used it to detect the object at angles $\{5^o, 15^o, \ldots 355^o\}$. The detection results were quite good, with a false alarm rate of 2.3 percent. When in error, the algorithm failed to distinguish between pairs of very similar objects, such as two toy cars that share a very similar appearance.

This example shows that the suggested algorithm has reasonable extrapolation capabilities, as it is trained on a relatively sparse set of images (rotations spaced 10^o apart).

On the average, six antiface detectors were sufficient to correctly detect the objects and 10 were required in the worst case. As noted before, the average time for classification of an N-pixel input image was less than $1.5N$ arithmetical operations. In terms of performance, while noting that we incurred a small percentage of false alarms, one should bear in mind that the problem addressed in this work is more general than in [13], [16], [20], which tackles only the problem of separating the COIL images from each other, as opposed to detection in general background setting (Fig. 12).

4 Detection under Varying Illumination

The antiface method can be extended to detection under varying illumination conditions by creating detectors which are insensitive to variation in the lighting. In this section, we briefly touch on this extension, using a simple illumination/shadow model. While the model is preliminary, hopefully it is adequate to explain the basic idea.

We use the well-known fact that the light reflected from a flat object with varying albedo can be described as

$$\mathbf{I}(\lambda) = \rho(\lambda)\mathbf{L}(\lambda), \tag{6}$$

where $\rho(\lambda)$ represents the reflectivity, $\mathbf{L}(\lambda)$ is the incident energy distribution, and λ is the wavelength. As in homomorphic filtering [7], the multiplicative nature of the

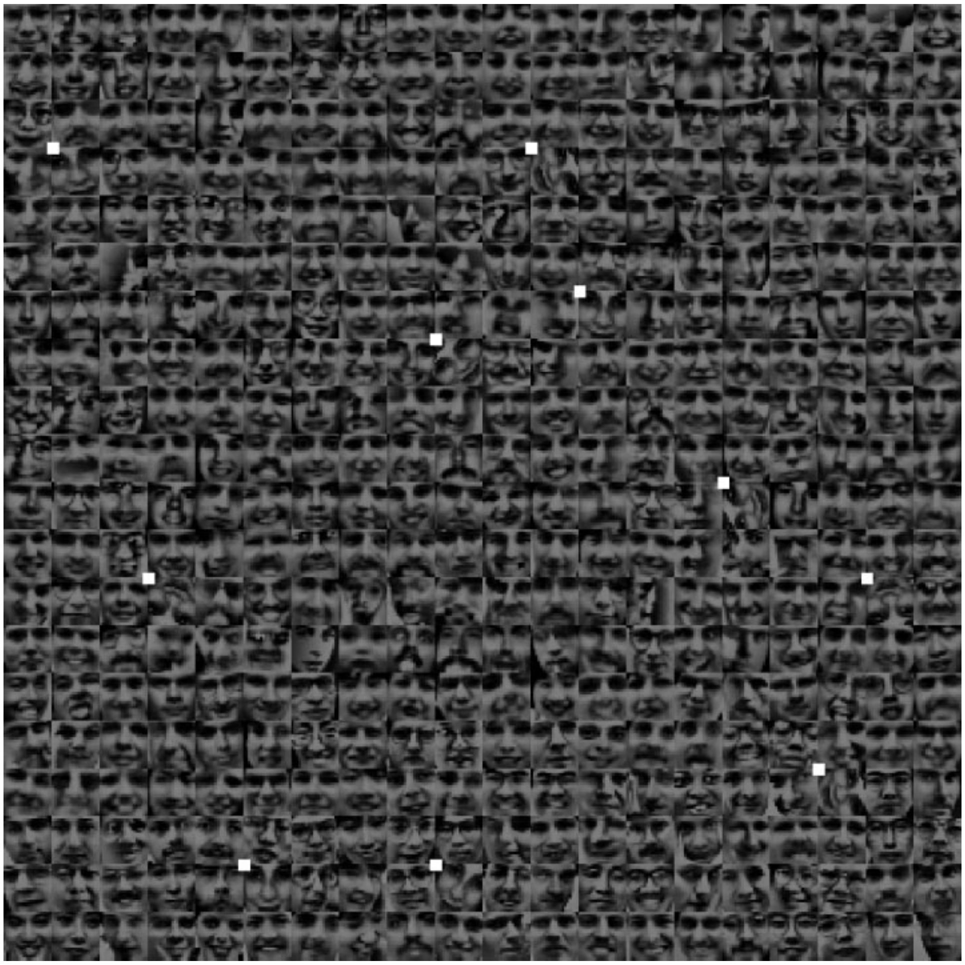

Fig. 8. Detection of "Esti" face, antiface method, four detectors; there are no false alarms, and exactly all the 10 "Esti" faces are detected. The fourth detector is applied only to subimages which passed the first, second, and third detector's test. The quality of results did not decrease when the background face images were also subject to linear transformations. The average number of detectors per pixel was 1.23.

reflectance model suggests applying a logarithm to (6), resulting in

$$\log(\mathbf{I}) = \log(\rho) + \log(\mathbf{L}),$$

where $\mathbf{I}$ is the given image, ρ represents the object's reflectance function, and $\mathbf{L}$ the lighting. Hence, to detect the object, we construct smooth detectors which operate in the logarithmic domain so that their inner products with ρ and $\mathbf{L}$ are small. This requires creating a training set which includes the images of the object to be detected, as well as images that model different light directions/conditions. Since average intensity differences are accounted for by normalizing the images, we chose to study some simple models corresponding to shadows being cast at different directions and positions, such as those depicted in Fig. 13. Since the shadows are part of the training set, the antiface method will detect them as multitemplate instances. To prevent this, a second, much smaller set of "shadow detectors" was created, using a multitemplate consisting of shadows. The detection process at first uses the detectors that were trained on the composite training set which includes object and shadow images and then it applies the shadow detectors only to the image regions detected by the first detector set. Those identified by the shadow detectors are removed, leaving only the instances of the multitemplate.

Fig. 14 presents some results. The multitemplate consists of 30×30 images of a planar object (a beer coaster), subject to arbitrary rotations. The shadow patterns used consisted of smooth step functions (see Fig. 13 for examples), with the following parameters:

1. The transition width of the step is $5 - 10$ pixels.
2. There is no restriction on the position/direction of the shadow in the image.

The detection in all experiments was successful. The number of multitemplate detectors ranged from three to eight and one to three shadow detectors were required.

5 Conclusions and Further Research

A novel detection algorithm—"antifaces"—was presented and successfully applied to detect various image classes of

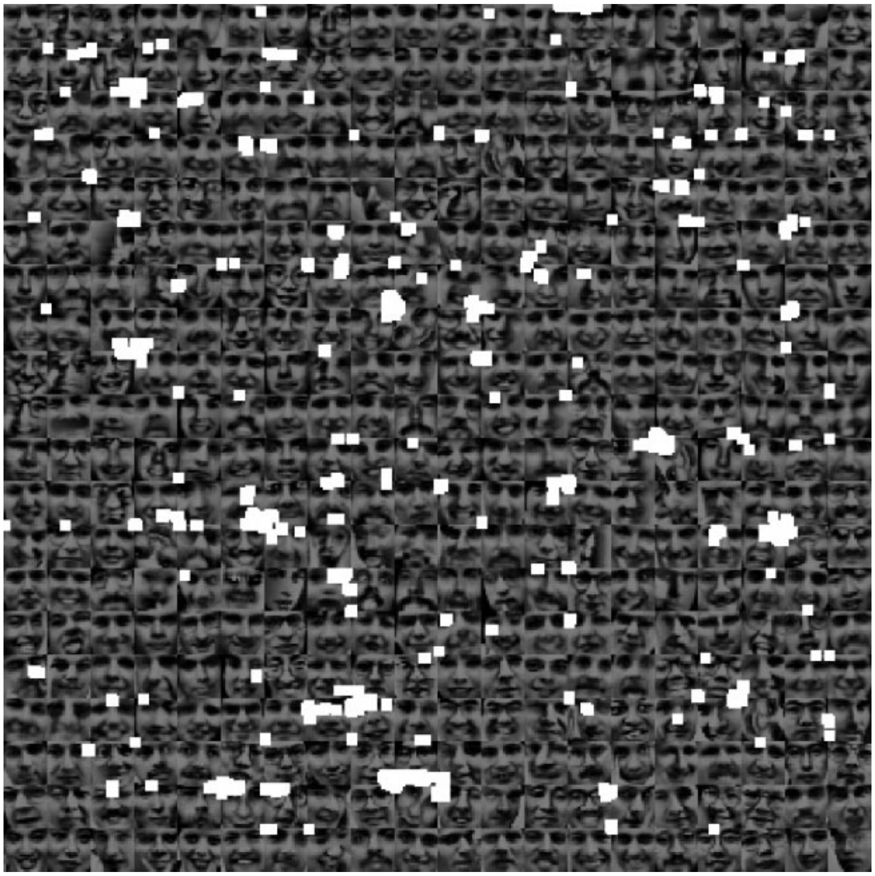

Fig. 9. Detection of "Esti" face, eigenface method. The multitemplate and the background are the same as those used in the test presented in Figs. 4, 5, 6, 7, and 8. The eigenface method required a face space of dimension 145, to detect all the "Esti" faces without false alarms (that is, to achieve a result as the one depicted in Fig. 8). Here, the result for the eigenface method with a face space of dimension 100 is presented. Note a large number of false alarms, some consisting of portions from different faces.

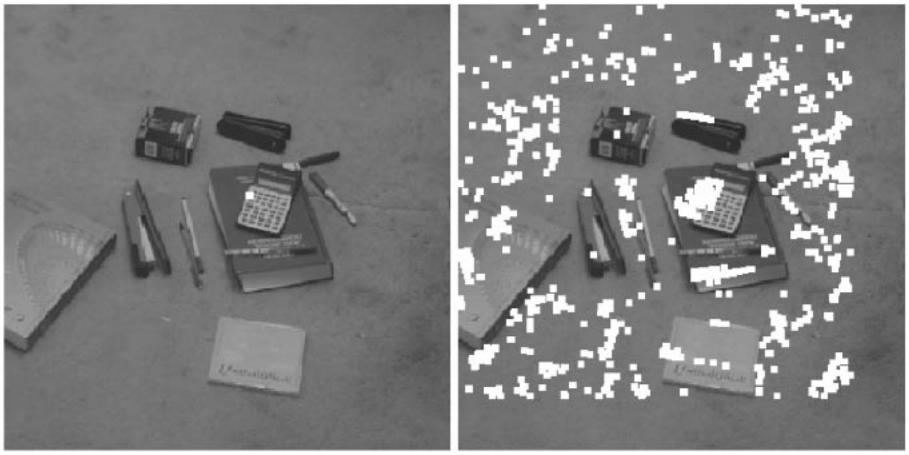

Fig. 10. An example of the detection of a pocket calculator in real images. Eight antiface detectors were sufficient to recover the calculator, without false alarms. A typical result is on the left. The eigenface method required a face space of dimension 30 to recover the calculator without false alarms. With a face space of dimension eight, there were many false alarms for the eigenface method (right).

the types which often occur in real-life problems. The case of varying illumination was also considered. The algorithm uses a simple observation on the statistics of natural images and a compact implicit representation of the image class to very quickly reduce false alarm rate in detection. In terms of speed, it is superior to both eigenface and support vector machine-based algorithms. No training on negative examples is required.

We plan to extend the antiface paradigm to other problems, such as the detection of 3D objects under a larger family of views, building a "generic" face detector, and event detection. Another direction that can be pursued is detection under uniform randomness.

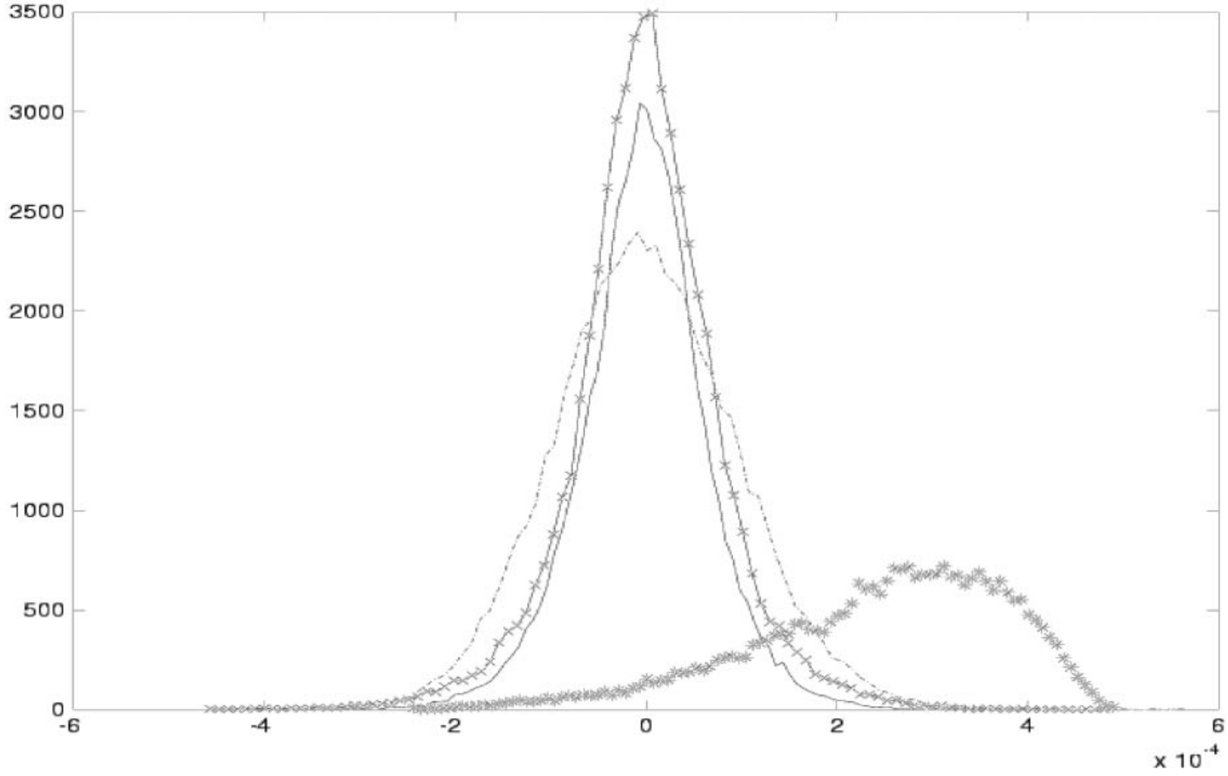

Fig. 11. Optimal separation of “Esti” multitemplate from random images, using the Fisher Linear Discriminant. The vertical axis stands for the number of images, the horizontal axis for the normalized projection. The projections of three sets of random images and the “Esti” set are shown. The “Esti” projections are depicted by asterisks, the random images’ projections are the three narrower Gaussian-like distributions. The middle one (plain line) corresponds to the random images used as a training set.

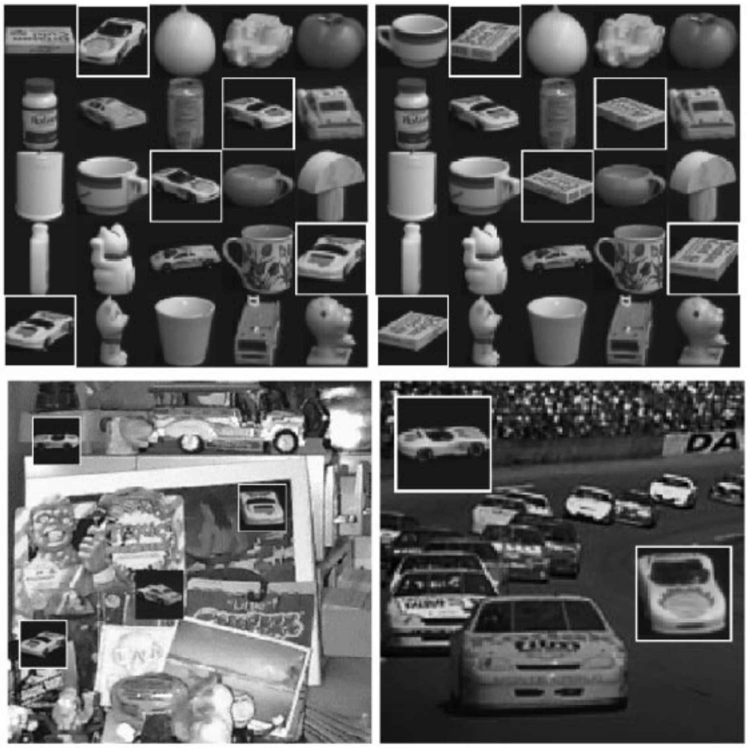

Fig. 12. Top: Detection of 3D objects from the COIL database: A toy car is sought in the left image; a chewing gum bar in the right. Bottom: Detection of 3D objects from the COIL database: A toy car is detected in general background setting. The car model in the center (left image) is different than the one being sought. Detection is marked by a white square around the detected image region. The average number of detectors per pixel was 1.31.

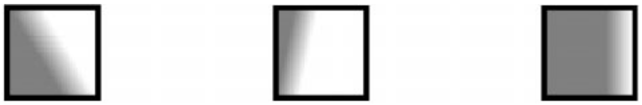

Fig. 13. Some simple shadow models.

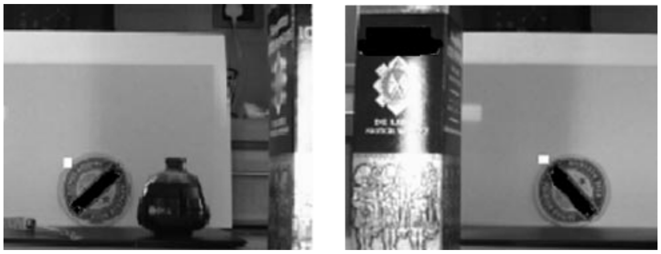

Fig. 14. Detection results for beer coaster under varying pose and illumination. The average number of detectors per pixel was 1.15.

Appendix

- Proof of (4). Since the exponential in the integrand factors into the product of exponentials of the form $\mathcal{I}^2(k,l)\exp(-(k^2+l^2)\mathcal{I}^2(k,l)d\mathcal{I}(k,l))$ and from symmetry considerations, it is enough to compute the one-dimensional integrals of the form

$$\int_{-\infty}^{\infty} t^2 \exp(-\alpha t^2)dt \propto \alpha^{-\frac{3}{2}}$$

where $\alpha > 0$. Equation (4) follows immediately from substituting $\alpha = k^2 + l^2$.
- Proof of (5) follows immediately from (4) and from noting that, for any inner product $<,>$, the following holds:

$$<\mathbf{x},\mathbf{y}> = \frac{<\mathbf{x}+\mathbf{y},\mathbf{x}+\mathbf{y}> - <\mathbf{x},\mathbf{x}> - <\mathbf{y},\mathbf{y}>}{2}.$$

Acknowledgments

This paper was substantially modified and extended following both reviews of an earlier submission to the ECCV 2000 conference and to *IEEE Transactions on Pattern Analysis and Machine Intelligence*. The authors are grateful to the reviewers for their insightful comments. The authors also thank Henry Rowley for providing the face images used for the experiments in Section 3.1. Part of this research was conducted while D. Keren and C. Gotsman were working for the Hewlett-Packard Company. This work was partially supported by the Israeli Ministry of Science under grant 1229.

References

[1] S. Baker and S.K. Nayar, "Pattern Rejection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 544-549, 1996.

[2] P.N. Belhumeur, P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 711-720, July 1997.

[3] J. Ben-Arie and K.R. Rao, "A Novel Approach for Template Matching by Nonorthogonal Image Expansion," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 3, no. 1, pp. 71-84, 1993.

[4] M. Bichsel and A.P. Pentland, "Human Face Recognition and the Face Image Set's Topology," *Proc. CVGIP: Image Understanding,* vol. 59, no. 2, pp. 254-261, 1994.

[5] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.

[6] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, no. 6, pp. 721-741, June 1984.

[7] R.C. Gonzalez and P.A. Wintz, *Digital Image Processing.* Addison-Wesley, 1992.

[8] D. Keren and M. Werman, "Probabilistic Analysis of Regularization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 10, pp. 982-995, Oct. 1993.

[9] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, pp. 103-108, 1990.

[10] Y. Lamdan and H.J. Wolfson, "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme," *Proc. Int'l. Conf. Computer Vision,* pp. 238-249, 1988.

[11] C.H. Lo and H.S. Don, "3D Moment Forms: Their Construction and Application to Object Identification and Positioning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, pp. 1053-1064, 1989.

[12] B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 696-710, July 1997.

[13] H. Murase and S.K. Nayar, "Visual Learning and Recognition of 3D Objects from Appearance," *Int'l J. Computer Vision,* vol. 14, no. 1, pp. 5-24, Jan. 1995.

[14] E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines: An Aapplication to Face Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1997.

[15] C.P. Papageorgiou, M. Oren, and T. Poggio, "A General Framework for Object Detection," *Proc. Int'l Conf. Computer Vision,* pp. 555-562, 1998.

[16] M. Pontil and A. Verri, "Support Vector Machines for 3D Object Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 6, pp. 637-646, June 1998.

[17] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes.* Cambridge Univ. Press, 1986.

[18] K.R. Rao and J. Ben-Arie, "Multiple Template Matching Using the Expansion Filter," *IEEE Trans. Video Technology,* vol. 4, no. 5, pp. 490-504, 1994.

[19] K.R. Rao and J. Ben-Arie, "Nonorthogonal Image Expansion Related to Optimal Template Matching in Complex Images," *Proc. CVGIP: Graphical Models and Image Processing,* vol. 56, no. 2, pp. 149-160, 1994.

[20] D. Roobaert and M.M. Van Hulle, "View-Based 3D Object Recognition with Support Vector Machines," *Proc. IEEE Int'l Workshop Neural Networks for Signal Processing,* pp. 77-84, 1999.

[21] H.A. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 1, pp. 23-38, 1998.

[22] H.A. Rowley, S. Baluja, and T. Kanade, "Rotation Invariant Neural Network-Based Face Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1998.

[23] A. Shashua, "On the Equivalence between the Support Vector Machine for Classification and Sparsified Fisher's Linear Discriminant," *Neural Processing Letters,* vol. 9, no. 2, pp. 129-139, 1999.

[24] L. Sirovich and M. Kirby, "Low-Dimensional Procedure for the Characterization of Human Faces," *J. Optical Soc. Am. A,* vol. 4, no. 3, pp. 519-524, 1987.

[25] G. Stockham, T.M. Cannon, and R.B. Ingebresten, "Blind Deconvolution through Digital Signal Processing," *Proc. IEEE,* vol. 63, pp. 678-692, 1975.

[26] K.K. Sung and T. Poggio, "Example-Based Learning for View-Based Human Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 1, pp. 39-51, Jan. 1998.

[27] D.L. Swets and J. Weng, "Using Discriminant Eigenfeatures for Image Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 8, pp. 831-836, Aug. 1996.

[28] D. Terzopoulos, "Regularization of Visual Problems Involving Discontinuities," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, no. 8, pp. 413-424, Aug. 1986.

[29] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience,* vol. 3, no. 1, pp. 71-86, 1991.

[30] M. Turk and A. Pentland, "Face Recognition Using Eigenfaces," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* pp. 586-591, 1991.

[31] M. Turk, personal communication, Dec. 1999.

[32] M. Uenohara and T. Kanade, "Use of the Fourier and Karhunen-Loeve Decomposition for Fast Pattern Matching with a Large Set of Templates," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 8, pp. 891-897, Aug. 1997.

[33] V.N. Vapnik, *The Nature of Statistical Learning Theory.* Berlin: Springer-Verlag, 1995.

[34] I. Weiss, "Geometric Invariants and Object Recognition," *Int'l J. Computer Vision,* vol. 10, no. 3, pp. 201-231, June 1993.

Daniel Keren received the PhD degree from the Hebrew University of Jerusalem and was a postdoctoral researcher at Brown University in the Division of Engineering. Since 1995, he has been with the Department of Computer Science at Haifa University, Haifa, Israel. He also consulted for three years at the Hewlett-Packard Company. His main research interests are computer vision, object modeling, and regularization.

Margarita Osadchy is currently pursuing the PhD degree in the Department of Computer Science, Haifa University, Haifa, Israel. Her main research interests are image and event detection and color image restoration.

Craig Gotsman received the PhD degree from the Hebrew University of Jerusalem. Since 1991, he has been on the Faculty of Computer Science at the Technion at Haifa, Israel. His main research interests are computer graphics, image rendering, and geometric modeling.