Anti-Faces for Detection

Daniel Keren[1], Margarita Osadchy[1], and Craig Gotsman[2]

[1] Department of Computer Science
Haifa University, Haifa 31905 Israel
Email: dkeren@cs.haifa.ac.il
[2] Faculty of Computer Science
Technion, Haifa 32000 Israel
Email: gotsman@cs.technion.ac.il

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 was also successfully applied to detect 3D objects under different views. Call the class of images, which should be detected, a *multi-template*.

The detection problem is solved by sequentially applying very simple filters (or *detectors*), which are designed to yield *small* results on the multi-template (hence "anti-faces"), 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, and by using a simple implicit representation of the multi-template.

Only images which passed the threshold test imposed by the first detector are examined by the second detector, etc. The detectors have the added bonus that they act independently, so that their false alarms are uncorrelated; this results in a percentage of false alarms which exponentially decreases in the number of detectors. This, in turn, leads to a very fast detection algorithm, usually requiring $(1 + \delta)N$ operations to classify an N-pixel image, where $\delta < 0.5$. Also, the algorithm requires no training loop.

The suggested algorithm's performance favorably compares to the well-known eigenface and support vector machine based algorithms, and it is substantially faster.

D. Vernon (Ed.): ECCV 2000, LNCS 1842, pp. 134–148, 2000.

1 Introduction

The well-known template detection problem in computer vision is: given a (usually small) image – the *template* – T, and a (usually much larger) image P, determine if 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 [16,2,12]. A more challenging problem is what we call *multi-template detection.* Here, we are given not one template T, but a *class* of templates $\mathcal{T}$ (which we call a *multi-template*), 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 m templates, we could apply the standard template detection techniques m times, and take the union of the results. This naive approach, however, breaks down in complexity for large m. The goal of this research is to develop an efficient algorithm for multi-template 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, or 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 multi-template 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

We proceed to define some relevant concepts, and outline the idea behind the detection scheme offered in this work. After surveying some related research, we lay the mathematical foundation for the anti-face algorithm. Following that, some experimental results are presented, and compared with eigenface and support vector machines based methods.

1.2 "Quick Rejection" vs. Detection and Recognition

The detection/recognition problem has a few stages, which converge to the solution. We term them as follows:

- The *quick rejection* stage: here, one tries to find a fast algorithm that filters out most input images that are not in the multi-template $\mathcal{T}$; usually, since we are searching a large image, the majority of input images – which, in this case, are the sub-images of the large image – will not be in $\mathcal{T}$. The quick rejection stage has to fulfill three requirements:

  1. It should be fast, due to the large amount of input images.
  2. It should not classify a member of $\mathcal{T}$ as a non-member.
  3. It should classify as little as possible non-members as members.

  It is well-known, for instance from the extensive research on the *eigenface* method [20,7] that, in some cases, $\mathcal{T}$ can be reasonably approximated by a linear subspace F, whose dimension k is quite smaller than the dimension of the ambient Euclidean space in which $\mathcal{T}$'s images reside. In the eigenface method, the quick rejection stage consists of casting aside the images whose distance from F is larger than a certain threshold, and it takes $O(k)$ convolutions to compute this distance [20]. However, in some cases k turns out to be quite large, as will be demonstrated in the sequel.
- The *detection* stage: here, one has to filter out the errors of the quick rejection stage – that is, detect the non-members of $\mathcal{T}$ which were not screened out in the quick rejection stage.
- The *recognition* stage: this optional stage consists of identifying and labeling $\mathcal{T}$'s members. For instance, after human faces have been detected in an image, one may wish to identify the corresponding individuals. However, there are cases in which recognition is not important or relevant; for instance, in detecting airplanes of a specific type, which are parked in an airfield, the recognition problem may be meaningless, since there is no distinction between airplanes parked at different angles.

This work addresses only the quick rejection stage. However, our empirical results were usually good enough to make the detection stage superfluous. Therefore we shall hereafter allow a slight abuse of terminology, by referring to the algorithm proposed here as a detection algorithm, and to the filters developed as *detectors*.

1.3 A Short Description of the Motivation Behind the Anti-Face Algorithm

The idea underlying the detection algorithm suggested here is very straightforward, and makes use of the fact that often an implicit representation is far more appropriate than an explicit one, for determining whether a certain element belongs to a given set.

Suppose, for instance, that we wish to determine whether a point in the plane, (x, y), belongs to the unit circle. Naturally, we will simply compute the value of $x^2 + y^2 - 1$; that is, we make use of the fact that there exists a simple functional, which assumes a value of zero on the set – and *only* on it. We could also use this fact to test whether a point is close to the unit circle. Let us term the idea of determining membership in a set $\mathcal{S}$, by using functionals which obtain a very small value on $\mathcal{S}$, as the *implicit approach*, and the functionals will be called *separating functionals* for $\mathcal{S}$. Note that this definition is more general than the classic "separating hyperplane" between classes; we are not trying to separate the set from its complement, or from some other set, by a hyperplane, or a quadratic surface etc., but characterize it by the range of values that certain functionals obtain on it. In general, this characterizes the set accepted by the detection process as semi-algebraic (if we restrict our functionals to be polynomials), or some other set, which is defined in the most general manner by:

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

where f_i are the separating functionals, and the ϵ_i are small. For the unit circle, for instance, one separating functional is required: $x^2 + y^2 - 1$.

So, to test whether a point (or, in our case, an image) x belongs to S, one has to verify that, for every $1 \leq i \leq m$, $|f_i(x)| \leq \epsilon_i$. The decision process can be shortened by first checking the condition for f_1, and applying f_2 only to the points (images) for which $|f_1(x)| \leq \epsilon_1$, etc.

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

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

Now, suppose one wishes to extend the implicit approach to the problem of quick rejection for a multi-template $\mathcal{T}$. Let us from here on replace "separating functional" by the more intuitive term *detector*.

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, we have to normalize the detectors, so assume that they are of unit length. If $|(d,t)|$ is very small for every $t \in \mathcal{T}$, then $f(y) = |(d,y)|$ is a candidate for a separating functional for $\mathcal{T}$. However, if we just choose such a few "random" d_i, this naive approach fails, as $|(d_i,y)|$ is very small also 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 axis. Thus, $\mathcal{T}$ consists of many projectively distorted images of the pocket calculator. Proceeding in a simplistic manner, 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, using a standard SVD decomposition of $\mathcal{T}$'s scatter matrix, and choosing the eigenvectors with the smallest eigenvalues. In Figs. 1-2 we show the result of this simplistic algorithm, which – not surprisingly – fails:

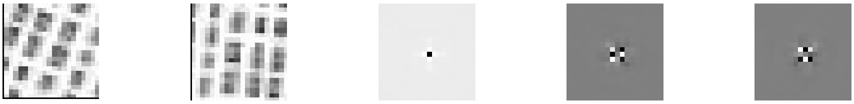

Fig. 1. Two of the members of the pocket calculator multi-template, and three of the "simplistic" detectors.

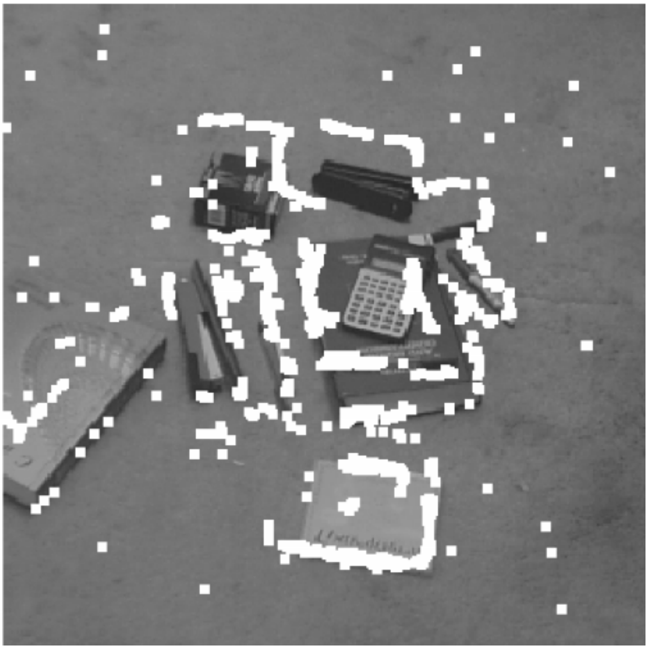

Fig. 2. The failure of the "simplistic" detectors, depicted in Fig. 1, to correctly locate the pocket calculator. Detection is marked by a small bright square at the upper left corner of the detected image region. Not only are there many false alarms, but the correct location is not detected. In the sequel, it will be shown that very accurate detection can be achieved by using better detectors.

Figs. 1-2 demonstrate that it is not enough for the detectors to yield small values on the multi-template $\mathcal{T}$; 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 sub-images 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 we will formally prove and quantify in the sequel, the absolute value of the inner product of two smooth vectors is large. If d is a candidate for a detector to the multi-template $\mathcal{T}$, suppose that not only is $|(d,t)|$ small for $t \in \mathcal{T}$, but also that d is smooth. Then, if $y \notin \mathcal{T}$, there is a high probability that $|(d,y)|$ will be large; this allows us to reject y, that is, determine that it is not a member of $\mathcal{T}$.

In the spirit of the prevailing terminology, we call such vectors d "anti-faces" (this does not mean that detection is restricted to human faces). Thus, a candidate image y will be rejected if, for some anti-face d, $|(d,y)|$ is larger than some 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 sub-images of a large image have to be tested, the first detector can be applied by convolution.

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 multi-template detection includes a large body of work on recognition of objects distorted under some geometric transformation group, using invariants [25]. Some intensity-based methods use moment invariants for recognition of objects under Euclidean or affine transformations [6]. 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 [5] have to contend with the considerable difficulty of locating features in the image. Methods that use differential invariants [25], 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 multi-template detection problem, the *eigenface* method of Turk and Pentland [20,21] has drawn a great deal of attention. This method approximates the multi-template $\mathcal{T}$ by a low-dimensional linear subspace F, usually called the *face space*. Images are classified as potential members of $\mathcal{T}$ if their distance from F is smaller than a certain threshold.

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 vs. "general" scatter [1,18,17], and a more elaborate modeling of the probability distribution in the face class [7]. In [8], eigenfaces were combined with a novel search technique to detect 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 have suggested 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 reaffirmed by one of the method's inventors [22].

In an attempt to apply the eigenface principle to detection under linear transformations [23], a version of the eigenface method is applied to detect an object with strong high-frequency components in a cluttered scene. However, the range of transformations was limited to rotation only, and only at the angles -50^0

to 50^0. The dimension of the face space used was 20. We will show results for a far more complicated family of transformations, using a number of detectors substantially smaller than 20.

Neural nets have been applied, with considerable success, to the problem of face detection [14], and also of faces under unknown rotation [15]. It is not clear whether the methods used in [15] 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 both 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) [11,9,10,13] are conceptually the method closest in spirit to the method suggested in this paper. An SVM consists of a function G which is applied to each candidate image t, and it classifies it as a member of the multi-template $\mathcal{T}$ or not, depending on the value of $G(t)$. A great deal of effort has been put into finding such a function which optimally characterizes $\mathcal{T}$. A typical choice is

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

where t is the image to be classified, x_i are the training images, y_i is 1 or -1 depending on whether x_i is in $\mathcal{T}$ (or a training set for $\mathcal{T}$) or not, and $K()$ a "classifier function" (for example, $K(t, x_i) = \exp(-\|t - x_i\|^2)$). Usually, a function is sought for which only a relatively small number of the x_i are used, and these 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 recovered by solving an optimization problem designed to yield a best separating hyperplane between $\mathcal{T}$ and its complement (or possibly between two different multi-templates). SVM's were introduced by Vapnik [24], and can be viewed as a mechanism to find the 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 t vectors, such as polynominals in these components, and allows for a more powerful detection scheme.

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 non-member of $\mathcal{T}$ is erroneously classified as a member).

3 The "Anti-Face" Method: Mathematical Foundation

To recap, for a multi-template $\mathcal{T}$, the "anti-face detectors" are defined as vectors satisfying the following three conditions:

- The absolute values of their inner product with $\mathcal{T}$'s images is small.
- They are as smooth as possible, so as to make the absolute values of their inner product with "random" images large; this is the characteristic which enables them to separate of $\mathcal{T}$'s images from random images. This will be formalized in Section 3.1.

- 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 more complicated condition. The independence of the detectors is crucial to the success of the algorithm, as it results in a number of false alarms which is exponentially decreasing in the number of detectors. This is explained in Section 3.2.

Once the detectors are found, the detection process is straightforward and 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. This allows a quick implementation using convolutions. 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 to detect not only the members of the training set (which is a sample of $\mathcal{T}$), but also images which are close to them, which suffices if the training set is dense enough in $\mathcal{T}$.

A schematic description of the detection algorithm is presented in Fig. 3.

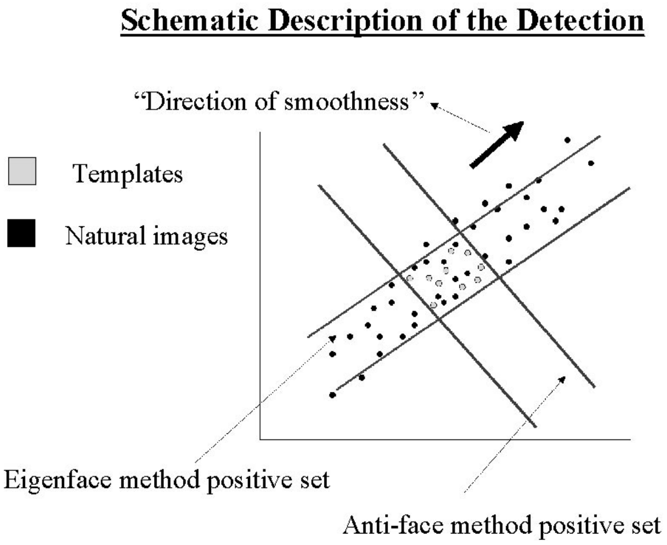
Schematic Description of the Detection
"Direction of smoothness"
□ Templates
■ Natural images
Eigenface method positive set
Anti-face method positive set

Fig. 3. Schematic description of the algorithm.

3.1 Computing the Expectation of the Inner Product

Let us 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 proved to be a reasonable model for natural images [4,3], assigns to an image I a probability proportional to the exponent of the negative of some "smoothness measure" for I. Usually, an expression such as $\iint (I_x^2 + I_y^2)dxdy$, or $\iint (I_{xx}^2 + 2I_{xy}^2 + I_{yy}^2)dxdy$, is used [4,19]. It is preferable, for the forthcoming analysis, to work in the frequency domain, since then the smoothness measure operator is diagonal, hence more manageable. The smoothness of a (normalized) $n \times n$ image I, denoted $S(I)$, is defined by

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

(note that is small for smooth images), and its probability is defined, following the Boltzman distribution, as

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

where $\mathcal{I}(k,l)$ are the DCT (Discrete Cosine Transform) coefficients of I. If 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 "random" in 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 F, of size $n \times n$, the expectation of the square of its inner product with a random image equals

$$E[(F,I)^2] = \int_{\mathcal{R}^{n\times n}} (F,I)^2 Pr(I)dI$$

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

$$\int_{\mathcal{R}^{n\times n-1}} (\sum_{(k,l)\neq(0,0)} \mathcal{F}(k,l)\mathcal{I}(k,l))^2 \exp(-\sum_{(k,l)\neq(0,0)}^{n} (k^2+l^2)\mathcal{I}(k,l))dI$$

which, after some manipulations, 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 Eq. 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 F be smooth.

This theoretical result is well-supported empirically. In Fig. 4, the empirical expectation of $(F,I)^2$ is plotted against Eq. 4. The expectation was computed for 5,000 different 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 Eq. 4 and the empirical expectation:

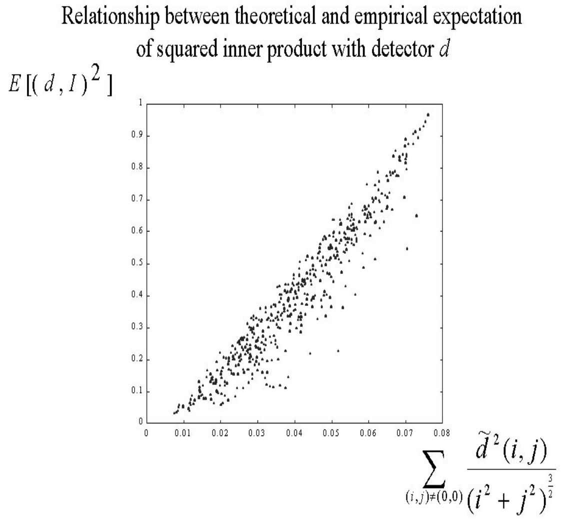

Fig. 4. Empirical verification of Eq. 4.

3.2 Forcing the Detectors to be Independent

It is difficult to expect that one detector can detect $\mathcal{T}$, without many false alarms. This is because, for a single detector d, although $|(d,y)|$ is large on the average for a random image y, there will always be many random images I such that $|(d,I)|$ 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*; that implies that if the false alarm rate (= percentage of false alarms) of d_1 is p_1, and that of d_2 is p_2, then the false alarm rate for both detectors will be p_1p_2. Since the entire detection scheme rests on the probability distribution defined in Eq. 3, the notion of independence is equivalent to the requirement that the two random variables, defined by $I \to (I,d_1)$ and $I \to (I,d_2)$, be independent, or

$$\int_{\mathcal{R}^{n\times n}} (I,d_1)(I,d_2)Pr(I)dI = 0$$

where $Pr(I)$ is defined as in Eq. 3. Denote this integral by $(d_1,d_2)^*$; it turns out to be

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

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

3.3 Finding the Detectors

To find the first anti-face detector, d_1, the following optimization problem is solved:

1. d_1 has to be of unit norm.
2. $|(d_1, t)|$ should be small, for every image t in the training set for the multi-template $\mathcal{T}$.
3. d_1 should be as smooth as possible under the first and second constraints, which will ensure that the expression in Eq. 4 will be large.

The solution we implemented proceeds as follows. First, choose an appropriate value for $\max\limits_{t \in \mathcal{T}} |(d_1, t)|$; 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 very rich – choose a larger M. Next, minimize

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

and, using a binary search on λ, set it so that $\max\limits_{t \in \mathcal{T}} |(d_1, t)| = M$.

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

We have also implemented a simpler algorithm, which minimizes the quadratic target function $\sum\limits_{t \in \mathcal{T}} (d_1, t)^2 + \lambda S(d_1)$. The resulting detectors are suboptimal, but usually 30% more such detectors will yield the same performance as the optimal ones.

4 Experimental Results

We have tested the anti-face method on both synthetic and real examples. In Section 4.1, it is compared against the eigenface method for the problem of detecting a frontal face subject to increasingly complicated families of transformations. In these experiments, the test images were synthetically created. In the other experiments, the anti-face 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 in various poses. Results for the COIL objects were compared with those of support vector machine based algorithms.

The results for the COIL database are not presented here, due to lack of space. Readers interested in a more complete version of this work, which includes these results, are welcome to mail the first author.

- A note on complexity: recall that each detector only tests the input images which have passed all the thresholds imposed by the preceding detectors. If, say, eight anti-face detectors were used, that does not mean that the number of operations for an input image with N pixels was $8N$. If, for example, the average false alarm rate for the detectors is 30%, then 70% of the input images will be discarded after the first detector, hence require N operations; of the 30% which pass the first detector, 21% will not pass the second detector, hence they'll require $2N$ operations, etc. Thus, the average number of operations per input image will be roughly $(0.7 \cdot 1 + 0.21 \cdot 2 + 0.063 \cdot 3 + ...)N$. In all our experiments, no more than $1.5N$ operations were required for classifying an N-pixel input image. Note that this analysis assumes that the large majority of input images are false alarms, a reasonable assumption if one searches all the sub-images of a large image.

4.1 Performance as Function of Multi-template's Complexity

In order to test the performance of the anti-face method with multi-templates of increasing complexity, we have created the following three multi-templates, each of which consists of a family of transformations applied to the frontal image of a face (20×20 pixels). The background consisted of other faces.

- Rotation only.
- Rotation and uniform scale at the range 0.7 to 1.3.
- The subgroup 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 multi-templates, we created the scatter matrix for a training set of each, and computed the number of largest eigenvalues whose sum equals 90% of the sum of all 400 eigenvalues. This is a rough measure of the "linear complexity" of the multi-template.

Ten images from each multi-template were then super-imposed on an image consisting of 400 human faces, each 20×20 pixels, and both the eigenface and anti-face algorithms were applied. These ten images were not in the training set.

Interestingly, while the eigenface method's performance decreased rapidly as the multi-template's complexity increased, there was hardly a decrease in the performance of the anti-face method. The next table summarized the results:

Algorithm's Performance	Rotation	Rotation + Scale	Linear
Number of Eigenvalues Required for 90% Energy	13	38	68
Eigenfaces 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

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

Detectors and Results Some of the images in the multi-template are now shown, as well as the first four detectors and the detection result of the anti-face method, and also the result of the eigenface method with a face space of dimension 100.

Fig. 5. Sample 20 × 20 pixel templates, and the first three anti-face detectors.

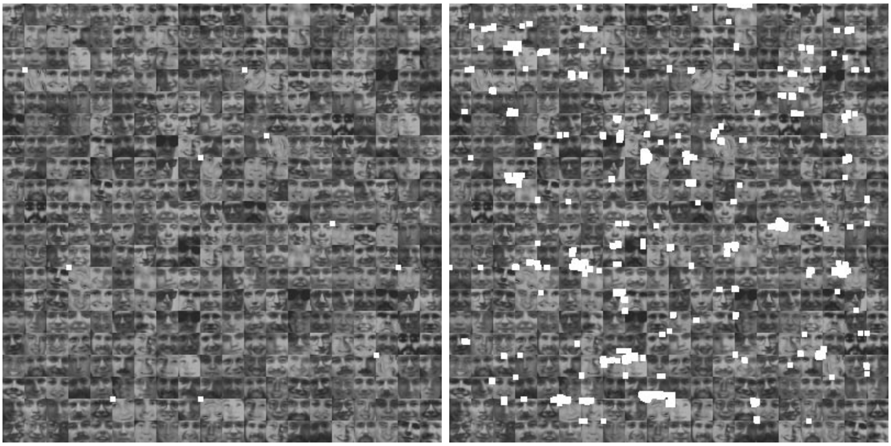

Fig. 6. Detection of “Esti” face, anti-face method (left), and eigenface method with a face space of dimension 100 (right).

4.2 Detection of Pocket Calculator

In this set of experiments, the problem of detecting a pocket calculator photographed from different angles and distances was tackled. Here, too, the anti-face method performed well, and eight detectors sufficed to recover the object in all the experiments without false alarms, which was substantially faster than the eigenface method.

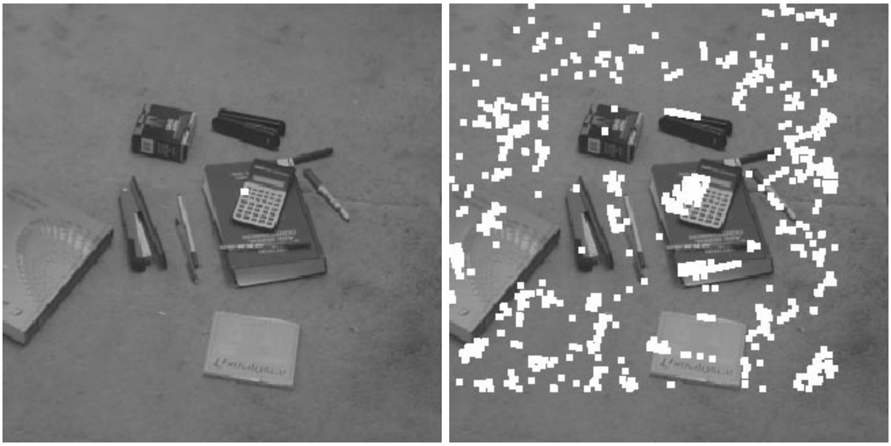

Fig. 7. Detection of pocket calculator, anti-face method (left), and eigenface method with a face space of dimension eight (right).

5 Conclusions and Further Research

A novel detection algorithm – "anti-faces" – was presented, and successfully applied to detect various image classes, of the type which often occur in real-life problems. 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 proved to be superior to both eigenface and support vector machine based algorithms.

We hope to extend the anti-face paradigm to other problems, such as detection of 3D objects under a larger family of views, and event detection.

References

1. P. N. Belhumeur, P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.
2. J. Ben-Arie and K.R. Rao. A novel approach for template matching by nonorthogonal image expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, 3(1):71–84, 1993.
3. S. Geman and D.Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:721–741, June 1984.
4. D. Keren and M. Werman. Probabilistic analysis of regularization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15:982–995, October 1993.
5. Y. Lamdan and H.J. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proc. Int'l. Conf. Comp. Vision*, pages 238–249, 1988.

6. C.H Lo and H.S. Don. 3-D moment forms: Their construction and application to object identification and positioning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11:1053–1064, 1989.
7. B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997.
8. H. Murase and S. K. Nayar. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.
9. E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.
10. C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *International Conference on Computer Vision*, pages 555–562, 1998.
11. M. Pontil and A. Verri. Support vector machines for 3d object recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(6):637–646, 1998.
12. K.R. Rao and J. Ben-Arie. Nonorthogonal image expansion related to optimal template matching in complex images. *CVGIP: Graphical Models and Image Processing*, 56(2):149–160, 1994.
13. D. Roobaert and M.M. Van Hulle. View-based 3d object recognition with support vector machines. In *IEEE International Workshop on Neural Networks for Signal Processing*, pages 77–84, USA, 1999.
14. H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.
15. H. A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.
16. G. Stockham, T.M. Cannon, and R.B. Ingebresten. Blind deconvolution through digital signal processing. *Proceedings of the IEEE*, 63:678–692, 1975.
17. K.K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.
18. D. L. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(8):831–836, 1996.
19. D. Terzopoulos. Regularization of visual problems involving discontinuities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8:413–424, August 1986.
20. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
21. M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proceedings of the Int'l Conf. on Computer Vision and Pattern Recognition*, pages 586–591, 1991.
22. Matthew Turk. *Personal communication*, December 1999.
23. 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. on Pattern Analysis and Machine Intelligence*, 19(8):891–897, 1997.
24. V. N. Vapnik. *The Nature of Statistical Learning Theory.* Berlin: Springer-Verlag, 1995.
25. I. Weiss. Geometric invariants and object recognition. *International Journal of Computer Vision*, 10:3:201–231, June 1993.