ARMY RESEARCH LABORATORY

Matching Pursuit Filters Applied to Face Identification

P. Jonathon Phillips

ARL-TR-1487

October 1998

19981106 040

Approved for public release; distribution unlimited.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Adelphi, MD 20783-1197

ARL-TR-1487 **October 1998**

Matching Pursuit Filters Applied to Face Identification

P. Jonathon Phillips

Sensors and Electron Devices Directorate

Sponsored by

DoD Counterdrug Technology Development Program Office

Naval Surface Warfare Center, Dahlgren Division B07
17320 Dahlgren Road
Dahlgren, VA 22448-5100

The material in this report also appeared as
P. Jonathon Phillips, *Matching Pursuit Filters Applied to Face Identification*, IEEE Trans. Image Process. **7**, No. 8 (August 1998), 1150–1164.

Approved for public release; distribution unlimited.

Abstract

A face identification algorithm is presented that automatically processes an unknown image by locating and identifying the face. The heart of the algorithm is the use of pursuit filters. A matching pursuit filter is an adapted wavelet expansion, where the expansion is adapted to both the data and the pattern recognition problem being addressed. For identification, the filters find the features that differentiate among faces, whereas for detection, the filters encode the similarities among faces. The filters are designed through a simultaneous decomposition of a training set into a two-dimensional wavelet expansion. This yields a representation that is explicitly two-dimensional and encodes information locally.

The algorithm uses coarse to fine processing to locate a small set of key facial features, which are restricted to the nose and eye regions of the face. The result is an algorithm that is robust to variations in facial expression, hair style, and the surrounding environment. Based on the locations of the facial features, the identification module searches the database for the identity of the unknown face using matching pursuit filters to make the identification.

The algorithm was demonstrated on three sets of images. The first set was images from the FERET database. The second set was infrared and visible images of the same people. (These two sets allowed the examination of algorithm performance on infrared and visible images individually, and on fused data from both modalities.) The third set of images was mugshot data from a law enforcement application.

Contents

1 Introduction **1**

2 Matching Pursuit Filters **6**

2.1 Matching Pursuit Filters for Detection 6

2.2 Identification 12

3 The Face Identification System **14**

3.1 Feature Location 16

3.2 Identification of Individuals 18

4 Experiments **20**

4.1 Decomposition of One Instance 20

4.2 Face Recognition from FERET Images 20

4.3 Face Recognition from Infrared and Visible Imagery 23

4.4 Mugshot Gallery 25

5 Conclusion **27**

Acknowledgments **29**

References **30**

Distribution **33**

Report Documentation Page **35**

Figures

1 Reconstruction of faces from an identification filter 4

2 A matching pursuit filter scanning an image 8

3 Pseudo-code for basis selection algorithm 9

4 Pseudo-code for evaluating C_g during design of detection filters 10

5 Design of detection filters 11

6 Pseudo-code for scanning an image with a detection filter . . 11

7 Pseudo-code for scanning an image with an identification filter 13

8 System organization 14

9 Facial features used 15

10 Pseudo-code for face detection and feature location algorithm 17

11 Pseudo-code for identification algorithm 19

12 Decomposition of one image 21

13 Face images from FERET database 22

14 Images of same person in infrared and visible 24

15 Performance of MPF and eigenfaces on mugshot gallery . . 26

Tables

1 Percentage of correctly identified faces for a database of 311 individuals 22

2 Performance as number of facial features is reduced 23

3 Comparison of recognition of infrared and visible images in a database of 101 individuals 24

4 Face recognition when data from infrared and visible images are fused 25

1. Introduction

There are many applications in modern society for a successful face identification system: nonintrusive identification and verification for credit cards and ATM machines, nonintrusive access control to buildings and restricted areas, and monitoring of ports of entry for terrorists and smugglers. For the designer of pattern recognition algorithms, face recognition is a very challenging problem. The goal is to develop an algorithm that can differentiate among a population of three-dimensional curved objects that all have the same basic shape, from databases whose sizes vary from a couple of hundred individuals to over one million. Further, the face itself is a dynamically varying object: facial expressions, makeup, facial hair, and hairstyle all change over time. The conditions under which facial imagery is collected also contribute to the difficulty of developing face recognition algorithms. The lighting, background, pose of the face, scale, and parameters of the acquisition are all variables in facial imagery collected under real-world scenarios. A key to successfully developing a general face identification system is to systematically solve a sequence of subproblems of increasing complexity. One critical subproblem is the development of an algorithm that can identify faces from a database of frontal facial imagery.

In this report, I describe an algorithm to perform the above task that is based on matching pursuit filters, a small set of facial features, and a simple geometric model of the face. The set of features consists of the nose and eye regions of the face and the interior of the face at a reduced scale. Since the nose and eye regions are the most stable and least varying parts of the face, restricting attention to these features increases the robustness of the algorithm with respect to variations in facial expressions, hair style, and background. The interior of the face is included so that information is encoded on the overall shape of the face. The geometric model describes the spatial relationship between the facial features: e.g., the eyes are above the nose, and the left eye is to the left of the right eye. The knowledge of the spatial relationship guides the search for facial features and ensures a realistic arrangement of features during identification.

Face recognition is substantially different from classical pattern recognition problems, such as character recognition. In character recognition there are a limited number of classes (usually less than 50), with a large number of training examples in each class, whereas in face recognition, there

are a large number of faces, or classes, and only a few training examples per face. In our case, there is only one training example per person, and the size of the gallery exceeds 300 individuals. (A *gallery* is a collection of images of known individuals; an image of an unknown face presented to the algorithm is called a *probe*.) Because of the size of the gallery, it is neither practical nor desirable to handcraft a representation that characterizes faces. Therefore, to be able to identify faces from large galleries, we need a method to automatically find features that distinguish one face from another.

The neural network community is pursuing techniques that automatically select features that distinguish among classes of objects. A few relevant techniques are feed-forward networks [1], principal component analysis [2], projection pursuit [3–5], factorial analysis [6,7], dynamic link architectures [8], and entropy-based wavelet encoding of images [9]. These techniques either originated in the neural network community (feed-forward networks, factorial analysis, and dynamic link architecture), or have found applications within neural networks (principal component analysis and projection pursuit). The work of a number of these authors [5–9] is motivated by theories of the human visual system. The goal of these approaches is to find representations that are data driven.

The heart of our face recognition algorithm is a new tool for creating efficient and compact models, called the *matching pursuit filter* technique, which is an adaptive wavelet expansion. A wavelet expansion of an image is adaptive if the choice of the wavelet basis depends on the image(s). The main innovation of a matching pursuit filter is that it is a wavelet expansion that is both data- and problem-adaptive; i.e., the expansion is adapted to both the data and the pattern recognition problem being addressed. This contrasts with most adaptive schemes, where the representation is a function of the data, but not a function of the problem to be solved. In a problem-driven expansion such as a matching pursuit filter, a filter that *detects* faces is designed to encode the *similarities* of all faces in the training set, whereas the filter that *identifies* faces automatically encodes the *differences* among the faces in the training set.

In image compression and signal analysis, adaptive wavelets have been used to decompose an individual image or signal [9–12]. In these works an algorithm selects the wavelets by minimizing a cost function: the error between the reconstructed image and the original image. Because a single image or signal is being decomposed, it is possible to find the optimal basis. The matching pursuit technique of Mallat and Zhang [11] uses a greedy algorithm to decompose an individual one-dimensional signal. In this work I generalize Mallat and Zhang's matching pursuit algorithm to simultaneously decompose multiple images for application to pattern recognition.

In the applications presented here, the matching pursuit filter design algorithm simultaneously decomposes all the images in the training set; thus it is not computationally feasible to optimize a global cost function. Instead, the algorithm selects the basis elements by using a greedy algorithm. In each iteration of the greedy algorithm, the algorithm chooses the next wavelet in the expansion by minimizing a cost function. For matching pursuit filters, the cost function incorporates the decision-making criterion of the associated pattern recognition problem.

The use of matching pursuit filters is a nonparametric technique for finding the differences among faces. Two related parametric techniques, which are cases of projection pursuit, are principal component analysis and Fisher discriminant analysis [13]. (Principal component analysis has been applied to face recognition [14–16] and to object recognition and detection [17,18]; discriminant analysis has been applied to face recognition [19–21].) Fisher discriminant analysis gives the optimal linear discriminant among classes (faces) when the distribution of each class is Gaussian. When there is only one example per class, discriminant analysis reduces to a variant of principal component analysis, which produces the optimal linear compression for least square error.

With one example per face, the assumption behind principal component analysis is that compression correlates with the differences among faces. In contrast, matching pursuit filters explicitly find these differences. (In this report, I concentrate on the case where there is one example per face.) For multiple examples per face, the matching pursuit expansion cost function could be modified to incorporate such information. Theoretically, differences in performance are determined by how well the distribution of the images of the face is modeled by a Gaussian distribution. The closer to a Gaussian distribution, the better the performance of Fisher discriminant analysis. For one example per face, I compare the performance of matching pursuit filters and principal component analysis in two experiments (sect. 4.3 and 4.4).

To exploit the spatial structure in images, matching pursuit filters explicitly model the two-dimensional structure of objects: the images are treated as functions in two variables, and the wavelet basis is constructed from two-dimensional directional filters. Because wavelets are local filters, one can directly see how the model relates to the training examples. As a result, the spatial arrangement of the wavelet basis has a tangible meaning in relation to objects in the training set. Figure 1 shows the reconstructions of four faces from an identification filter. By examining the reconstructions of the faces, one sees how the filter encodes facial differences. The reconstructions are not faithful to the original images, because the filter design algorithm

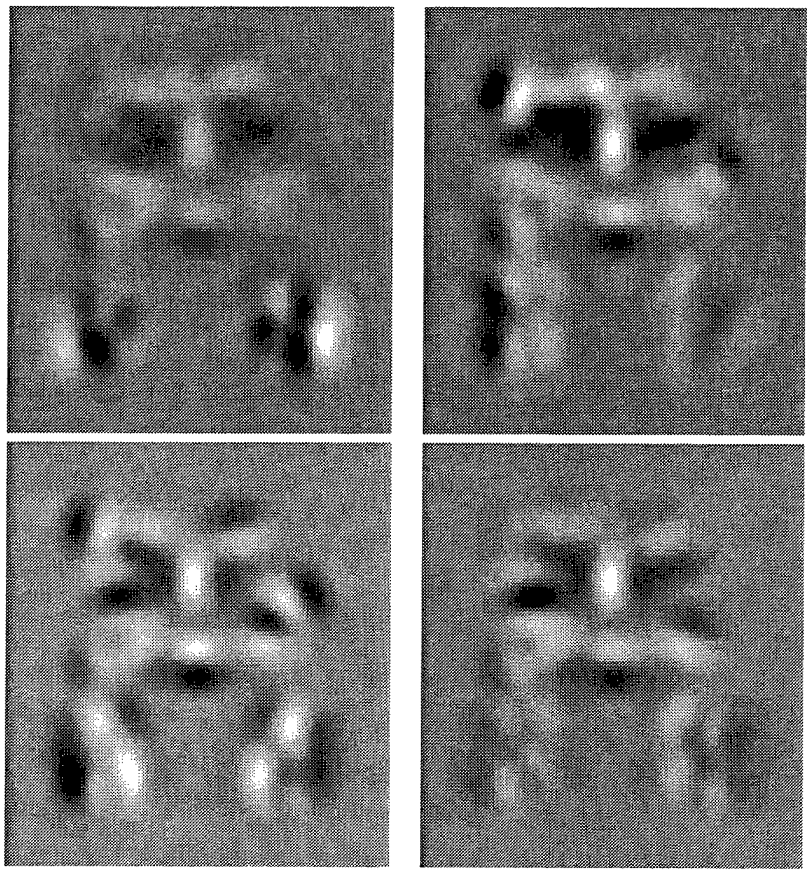


Figure 1. Reconstructions of faces from an identification filter. Images are reconstructed with equation (5), sect. 2.1, with 30 coefficients.

selects a basis that differentiates among faces, not a basis that minimizes the reconstruction error. In matching pursuit filters, the explicit representation of the image by two-dimensional wavelets captures both local and global features. This contrasts with discriminant analysis and principal component analysis, where images are treated as vectors, and the representation is global.

In computer vision, artificial intelligence, and neuroscience, approaches to finding representations of objects from data are active areas of research. The dynamic link architecture of Lades *et al.* [8] is a general object recognition technique that represents objects by projecting an image onto a rectangular array of Gabor jets. Wiskott *et al.* [22] have specialized the architecture for face recognition. The work of Rao and Ballard on iconic representation [23,24] takes a similar approach, except that images are projected onto a jet of directional derivatives of Gaussian densities. In dynamic link architecture and iconic representation, the basis (filters) that represents an object

is chosen by the algorithm designer. In the work of Viola [25], a basis is iteratively constructed from subwindows in a set of training images. The selection criterion for basis vectors is their orthogonality to basis vectors already chosen, not their power to separate classes.

Two measures of success for a face recognition algorithm are the ability to recognize faces from a large gallery and the ability to automatically process probes. In the algorithm represented here, images are automatically processed with a two-stage system. The first stage locates the face and a small set of facial features. The second stage identifies the face given the location from the first stage. In this report, I demonstrate algorithm performance by identifying faces from two large galleries (sect. 4.2 and 4.4). The first is a gallery of 311 individuals, with one image per person; the images are from the FERET database [26]. The second is a gallery of 2175 individuals, with images from mugshot data.

This approach contrasts with the majority of algorithms described in the literature, for which results are given on small galleries (<50 individuals) with many images per person in the gallery (>5 images). Since these algorithms also require that the face be in a predetermined position, the first stage is not needed.

Only a handful of algorithms have been tested on galleries of more than 150 individuals and have the ability to process images automatically [22,27–30]. Swets and Weng [31] tested their algorithm on FERET and other images, but their algorithm is not fully automatic. Cox *et al.* [32] identified faces from a database of 685 images, but required that an operator manually locate 35 points on each face.

One area of recent interest is recognizing faces in infrared (IR) imagery. This interest is driven by the ability of IR cameras to acquire images of faces in the dark. This report also presents results from a study comparing algorithm performance on visible versus IR images for face recognition.

2. Matching Pursuit Filters

The original matching pursuit idea of Mallat and Zhang [11] uses a greedy heuristic to iteratively construct a best-adapted decomposition of a function f on $\Re$. The algorithm works by choosing at each iteration i the wavelet g in the dictionary $\mathcal{D}$ that has maximal projections onto the residue of f. The best-adapted decomposition is selected by the following greedy strategy. Let $R^0 f = f$; then g_i is chosen such that

$$|\langle R^i f, g_i \rangle| = \max_{g \in \mathcal{D}} |\langle R^i f, g_i \rangle|, \tag{1}$$

where

$$R^{i+1} f = R^i f - \langle R^i f, g_i \rangle g_i$$

for $i \geq 1$.

The algorithm selects each wavelet in the expansion by maximizing the right-hand term in equation (1). This equation allows for an expansion based on a single function and minimizes the reconstruction error. To extend the technique to pattern recognition, we replace the right-hand side with a function C_g, which (1) allows for the simultaneous expansion of multiple templates (functions), and (2) incorporates knowledge of the pattern recognition problem being addressed. The extension from functions f on $\Re$ to functions (templates) t on $\Re^2$ is straightforward: a dictionary of two-dimensional wavelets is used.

2.1 Matching Pursuit Filters for Detection

Matching pursuit filters have two components. The first component is how the face or a facial feature is represented—for example, all noses. For matching pursuit filters, all noses are represented by a given basis. The second component is the representation of a particular nose. I first discuss the encoding of an instance of a nose for a given basis, and then show how the basis is selected.

A particular nose (or in the more general case, an instance of an object) is represented as an n-dimensional vector $(a_0, \ldots, a_{n-1})$, called a coefficient vector. One computes the coefficient values a_i by projecting the image of a nose onto a basis $\{g_0, \ldots, g_{n-1}\}$, which need not be orthogonal. Because the basis is not necessarily orthogonal, an iterative projection algorithm calculates the coefficients. If the basis is orthogonal, then the algorithm reduces

to the standard projection method. The projection algorithm adjusts for the nonorthogonality by using residual images. If t is an image or template, then $R^i t$ is the residual image during iteration i, where $R^0 t = t$. The coefficient a_i is the projection of the residual image R^i onto the basis element g_i, or mathematically,

$$a_i = \langle R^i t, g_i \rangle, \tag{2}$$

where $\langle \cdot, \cdot \rangle$ is the inner product between two functions. The residual image is updated after each iteration by

$$R^i t = R^{i-1} t - a_{i-1} g_{i-1}, \tag{3}$$

for $i \geq 1$.

After the nth iteration, an image t is decomposed into a sum of residual images:

$$t = \sum_{i=0}^{n-1} (R^i t - R^{i+1} t) + R^n t. \tag{4}$$

Rearranging equation (3) and substituting into equation (4) yields

$$t = \sum_{i=0}^{n-1} a_i g_i + R^n t,$$

and the approximation of the original image after n iterations is given by

$$\widehat{t} = \sum_{i=0}^{n-1} a_i g_i. \tag{5}$$

The approximation need not be very accurate, because only enough information is encoded to allow detection. By examining the reconstruction $\widehat{t}$, one sees which features or details distinguish noses from non-noses. In contrast, reconstructions from adapted wavelet expansions based on an image compression admissibility criterion will have a greater fidelity to the original image(s).

The goal of the detection algorithm is to determine whether an observed pattern belongs to a particular class: i.e., is this a nose? For this determination to be made, there must be a way of measuring the similarity between two objects or patterns. With matching pursuit filters, one compares the coefficient vectors from two objects, where the coefficient vectors are generated by the same basis. The similarity measure between two objects is the angle between their coefficient vectors. This measure is invariant under linear changes in the contrast of the image. Furthermore, if the basis is composed of wavelets, then the similarity measure is also invariant to the illumination level in the image. An L_2 function t is a wavelet if $\int t(\mathbf{x})\, d\mathbf{x} = 0$;

this requirement is referred to as the wavelet condition [33]. More precisely, the similarity measure is invariant to linear changes in illumination.

The second part of representing noses is choosing an appropriate basis. In the ideal basis, all noses would have the same coefficient vector, and all occurrences of this coefficient vector would be a nose. Unfortunately, this does not occur. An alternative criterion is to select a basis where the coefficient vectors that represent noses cluster. The vector that is the cluster center is referred to as a *proto-nose* (or, in general, a *proto-object*). This vector, or coefficient vector, represents an average nose. The matching pursuit filter design algorithm searches for such a representation.

The matching pursuit filter is trained on m different examples of noses. Let $\{t_1, \ldots, t_m\}$ be m examples of noses, where t_l contains one example of a nose. The noses are aligned in the templates so that the center of the nose is the origin. For objects other than noses, the examples are aligned about a common point. Using these examples, the algorithm selects the basis elements from a dictionary $\mathcal{D}$.

In the work described here, a dictionary is composed of two-dimensional directional wavelets. These wavelets were chosen because they encode information locally at different scales and orientations. The basis elements in the dictionary do not span the space of possible images. The dictionary excludes high-frequency wavelets to reduce the effect of high-frequency noise. It also excludes low-frequency wavelets, for computational considerations and to avoid encoding information in the background. The wavelets in these dictionaries can be centered at any place in the region containing a nose in the training set (fig. 2).

For face recognition, the algorithm uses a dictionary derived from the second partial derivatives of Gaussian densities and their Hilbert transforms, which were selected because they are directional edge detectors. The wavelets do not need to be self-inverting because I am not interested in recon-

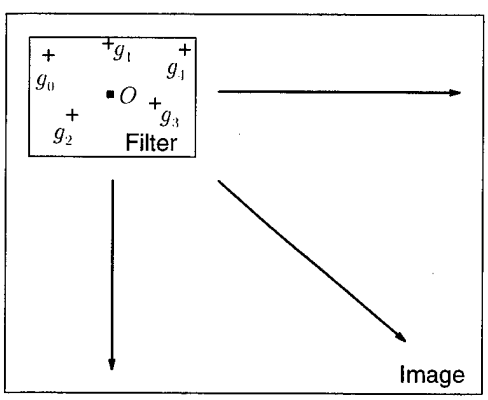


Figure 2. A matching pursuit filter scanning an image. Center of filter is O, which moves as image is scanned. This filter has five basis elements, g_0, g_1, g_2, g_3, and g_4. Centers of wavelets g_i relative to O are marked by "+" signs.

structing the images. (I do not address the issue of what an "optimal" dictionary is for a particular problem. This difficult problem is beyond the scope of this report.)

A greedy algorithm selects the basis elements. In iteration i, the basis function g_i is selected. The choice of g_i is a function of the residual images $R^i t_l$ and coefficients α_j^l from previous iterations, i.e., $j < i$. Let the coefficient $\alpha_j^l = \langle R^j t_l, g_j \rangle$, that is, the jth coefficient for template l. The set of coefficients generated through the ith iteration is denoted by $\Lambda_i = \cup_l(\alpha_0^l, \ldots, \alpha_i^l)$, $i \geq 0$, and $\Lambda_{-1} = \emptyset$.

Each iteration of the basis selection algorithm consists of three steps. (Pseudo-code for the basis selection algorithm is given in fig. 3.) In the first step, the basis function g_i is selected. In the second step, the coefficient vectors for each template t_l are updated. In the third step, the residual images are updated by $R^{i+1} t_l = R^i t_l - \alpha_i^l g_i$. The ith basis function is selected by the following optimization procedure:

$$g_i = \arg\min_{g \in \mathcal{D}} C_g(R^i t_1, \ldots, R^i t_m, \Lambda_{i-1}),$$

where C_g measures how well the coefficient vectors cluster when the ith basis function is g. The function C_g is evaluated for each $g \in \mathcal{D}$, and the g that minimizes C_g is selected as the basis element g_i. (Pseudo-code for evaluating C_g is presented in fig. 4.) In the current implementation of C_g for a given g, the cluster is the mean of $(\alpha_0^l, \ldots, \alpha_{i-1}^l, \langle R^i t_l, g \rangle)$, $1 \leq l \leq m$. Once the cluster vector is determined, C_g computes the average distance from the coefficient vectors to the cluster vector. This distance is a measure of scatter (variance) of the coefficient vectors about the cluster vector. If the dispersion is small, then g is a good candidate for g_i; on the other hand, if the dispersion is large, then g is a poor choice. The technique extends to k proto-noses or cluster vectors. In this case, C_g generates k clusters and

Figure 3. Pseudo-code for basis selection algorithm.

```
1: R^0 t_l = t_l;
2: do i = 0 to i < number of iterations, n;
      \*number of iterations = number of desired basis elements.*\
      3: Compute C_g for each wavelet in the dictionary;
      4: Select the ith basis element,
         g_i is the wavelet that minimizes C_g;
      5: Update coefficients for each template,
         α_i^l = ⟨R^i t_l, g_i⟩;
      6: Update the residue images for each template ,
         R^{i+1} t_l = R^i t_l − α_i^l g_i;
      7: Increment the iteration counter,
         i = i + 1;
8: end do
```

Figure 4. Pseudo-code for evaluating C_g during design of detection filters.

g is the wavelet for which C_g is to be evaluated.
Iteration i.
$(\alpha_0^l, \ldots, \alpha_{i-1}^l)$ are the coefficients computed through iteration $i-1$ to represent template t_l.

1: Compute the centroid of $(\alpha_0^l, \ldots, \alpha_{i-1}^l, \langle R^i t_l, g \rangle)$;
Let μ be the centroid.
2: Compute the mean distance U between μ and $(\alpha_0^l, \ldots, \alpha_{i-1}^l, \langle R^i t_l, g \rangle)$,
$U = 1/M \sum \| \mu - (\alpha_0^l, \ldots, \alpha_{i-1}^l, \langle R^i t_l, g \rangle) \|$
3: Return U as the value of C_g;

measures the spread of the coefficient vectors from the cluster vectors. The clusters are found with a k-means algorithm. Figure 5 illustrates the design of a detection filter. Figures 5(a) to (d) are the training set, and 5(e) is the nose in 5(b), reconstructed by equation (5) with 30 coefficients.

The algorithm is iterated until n basis elements are selected. The choice of the number of basis elements depends on the performance level desired and is usually determined experimentally. If n is too small, then the false-alarm rate is too high; if n is too large, the filter will not generalize to noses outside the training set.

The output from the matching pursuit filter design algorithm is an ordered list of n basis elements and a list of n coefficients. The combination of both lists is a matching pursuit filter. If the filter design algorithm generates k proto-noses, the matching pursuit filter consists of the basis elements and the k coefficient lists (coefficient vectors). The location of the basis elements encodes the geometric structure of an object (fig. 2). The centers of the basis elements g_i are usually not aligned. This is illustrated in figure 1, where a filter represents an object (nose) that is larger than the support of an individual basis element.

A matching pursuit filter detects a nose by scanning a nose detection filter across an image (fig. 2), which results in a response image $\mathcal{F}$. The response at pixel (u_1, u_2) measures the similarity between the region centered at (u_1, u_2) and the proto-nose. One criterion detects the center of the nose at the maximum response in $\mathcal{F}$. An alternative method reports all points above a threshold as the center of the nose. The algorithm computes the response at a pixel (u_1, u_2) by comparing the proto-nose coefficient vector with an image coefficient vector $\mathbf{a}(u_1, u_2)$. There is an image coefficient vector $\mathbf{a}(u_1, u_2)$ for each pixel. (Fig. 6 is a pseudo-code description of the filter detection algorithm.)

Figure 5. Design of detection filters: (a) to (d) Training set. (e) Nose in (b) reconstructed by equation (5).

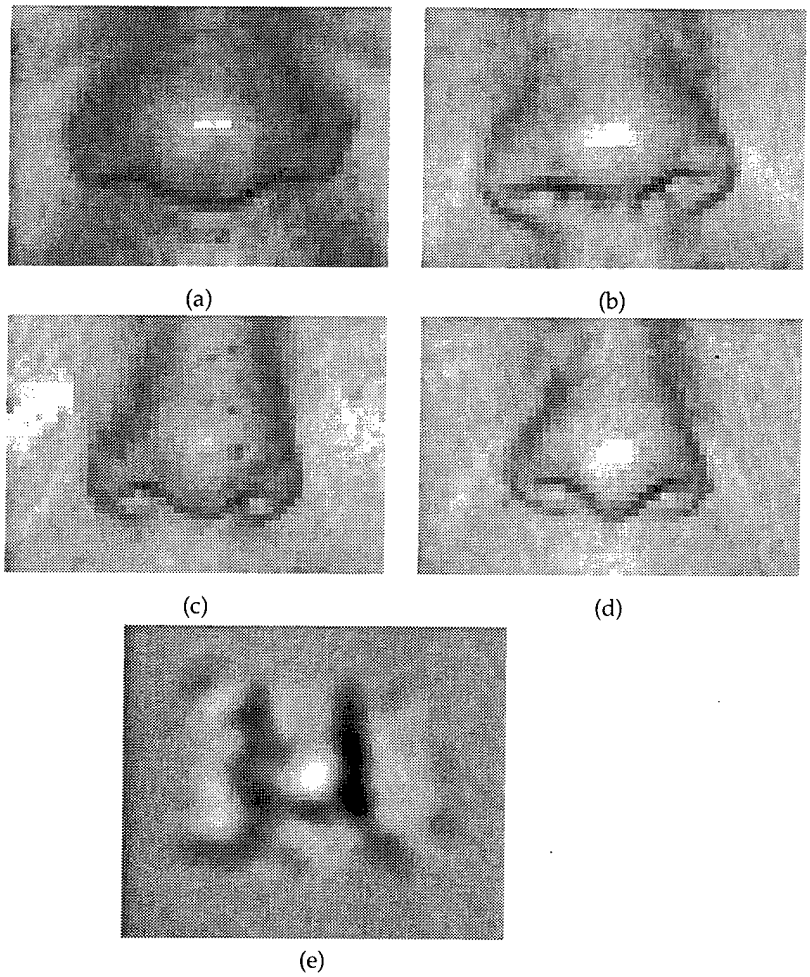


Figure 6. Pseudo-code for scanning an image with a detection filter.

$\mathcal{I}(u_1, u_2)$: input image.
$\Lambda = (\alpha_0, \ldots, \alpha_{n-1})$: coefficient vector that represents proto-nose.
$\mathcal{F}(u_1, u_2)$: response image.

1: **do** for all pixels (u_1, u_2) in image $\mathcal{I}$;
  2: Compute image coefficient vector $\mathbf{a}(u_1, u_2)$ using equation (6),
    $\mathbf{a}(u_1, u_2) = (a_0(u_1, u_2), \ldots, a_{n-1}(u_1, u_2))$;
  3: Compute filter response at pixel (u_1, u_2),
    $\mathcal{F}(u_1, u_2) = d_\theta(\Lambda, \mathbf{a}(u_1, u_2))$;
4: **end do**;
5: Search $\mathcal{F}$ for the location the nose(s);

The algorithm computes the image coefficient vector $\mathbf{a}(u_1, u_2)$ by expanding the image about the pixel (u_1, u_2). This expansion is accomplished by

translation of the basis elements g_i by (u_1, u_2), and projection of the image on the translated basis elements. Let $a_i(u_1, u_2)$ be the ith coefficient of $\mathbf{a}(u_1, u_2)$; then

$$a_i(u_1, u_2) = \langle R^i \mathcal{I}, g_i(\cdot + u_1, \cdot + u_2) \rangle. \tag{6}$$

After the image coefficient vectors have been determined, the next step computes the response image. Let $\Lambda = (\alpha_0, \ldots, \alpha_{n-1})$ be the cluster vector that represents the proto-nose; then $\mathcal{F}(u_1, u_2) = d_\theta(\Lambda, \mathbf{a}(u_1, u_2))$, where $d_\theta(\cdot, \cdot)$ is the cosine of the angle between two vectors; i.e., the response is the cosine of the angle between Λ and $\mathbf{a}(u_1, u_2)$. The last step searches $\mathcal{F}$ for noses. In some applications, the matching pursuit filter consists of more than one cluster coefficient vector. In this case, the response of the filter at each pixel is the maximum value of d_θ taken over all cluster vectors.

2.2 Identification

In detection problems such as those discussed in this work, we are interested in locating a face or facial feature, whereas in identification we are interested in distinguishing among faces. The filters are designed from images in the gallery, and the filters are used to identify unknown faces in probes. The generalization to more complex cases is straightforward; these include problems such as character recognition, where there is more than one training example per class. The overall strategy for designing matching pursuit filters for identification is the same as for detection, except that a different criterion is used to select the basis.

For detection, the matching pursuit filter design procedure selects a basis in which the coefficient vectors clustered, and only one coefficient vector Λ represents a class of objects. For detection, Λ is compared to image coefficient vectors. Because there is a single class, only one coefficient vector is needed. However, for identification, to distinguish among all the people in the database, there is a coefficient vector for each individual. Person l is represented by coefficient vector $\Lambda^l = \{\alpha_0^l, \ldots, \alpha_{n-1}^l\}$. To measure the similarity between an unknown face and individual l, we compare coefficient vectors $\mathbf{a}(u_1, u_2)$ and Λ^l.

A face centered at (u_1, u_2) is identified as person l if the distance between $\mathbf{a}(u_1, u_2)$ and Λ^l is minimized. To decrease the likelihood that faces are misidentified, the matching pursuit design algorithm searches for a basis that separates the Λ^l coefficient vectors. The algorithm for selecting the ith basis element for identification has the same three steps as for detection, but with a different function C_g. For identification,

$$C_g(R^i t_1, \ldots, R^i t_m, \Lambda_{i-1}) = -\sum_k \max_{l \neq k} d_\theta(k, l) + \lambda \sum_k \|(\alpha_0^k, \ldots, \alpha_{i-2}^k, \langle R^{i-1} t_k, g \rangle)\|$$

selects the ith basis function. The function $d_\theta(k,l)$ equals the cosine of the angle between $(\alpha_0^k,\ldots,\alpha_{i-2}^k,\langle R^{i-1}t_k,g\rangle)$ and $(\alpha_0^l,\ldots,\alpha_{i-2}^l,\langle R^{i-1}t_l,g\rangle)$. The coefficient vector $(\alpha_0^k,\ldots,\alpha_{i-2}^k)$ represents person k after the $i-1$th iteration. If g were selected for g_i, then $(\alpha_0^k,\ldots,\alpha_{i-2}^k,\ \langle R^{i-1}t_k,g\rangle)$ would represent person k after iteration i. The first term in C_g forces the coefficient vectors to separate, and the second term searches for sets of coefficient vectors with the largest average magnitude. The parameter λ sets the relative importance of the two terms. If the second term is not included, the filter becomes too sensitive to patterns in the background. Displayed in figure 1 are the reconstructions of four faces from an identification filter using equation (5). For identification, the output from the matching pursuit filter design algorithm is a list of n basis elements and a coefficient vector for each person in the training set. The procedure for identifying faces in images is a variant of the method that detects noses (fig. 7). As before, $\mathbf{a}(u_1,u_2)$ is the image coefficient vector centered at (u_1,u_2). For detection, a single response image $\mathcal{F}$ was computed; however, for identification, a response $\mathcal{F}^k$ is computed for each coefficient vector Λ^k, where $\mathcal{F}^k(u_1,u_2)=d_\theta(\Lambda^k,\mathbf{a}(u_1,u_2))$. The estimated identity of the person in the image is $\widehat{k}$, which is found by a search for the maximum response over all the $\mathcal{F}^k$ images. More precisely, the face in the image is identified as the person $\widehat{k}$ such that

$$\mathcal{F}^{\widehat{k}}(\widehat{u}_1,\widehat{u}_2)=\max_{k,(u_1,u_2)}\mathcal{F}^k(u_1,u_2),$$

where $(\widehat{u}_1,\widehat{u}_2)$ is the estimated center of the face in the image.

The extension to distinguishing among multiple classes with more than one example is straightforward. It is a combination of the detection and identification modes of designing matching pursuit filters. Each class is represented as a proto-object or cluster vector. The function C_g selects a basis that separates the cluster vectors while simultaneously forming clusters for like objects or classes.

Figure 7. Pseudo-code for scanning an image with an identification filter. (Goal is to estimate k.)

$\mathcal{I}(u_1,u_2)$: input image.
$\Lambda^k=(\alpha_0^k,\ldots,\alpha_{n-1}^k)$: coefficient vector that represents person k.
$\mathcal{F}^k(u_1,u_2)$: response of image $\mathcal{I}$ to Λ^k.
$\widehat{k}$: estimated identity of person in the image.

1: Compute $\mathbf{a}(u_1,u_2)$ for each pixel in $\mathcal{I}$;
2: Compute $\mathcal{F}^k(u_1,u_2)$,
  $\mathcal{F}^k(u_1,u_2)=d_\theta(\Lambda^k,\mathbf{a}(u_1,u_2))$;
3: Identify the person in image $\mathcal{I}$;
  Find $\widehat{k}$ and $(\widehat{u}_1,\widehat{u}_2)$,
  where $\mathcal{F}^{\widehat{k}}(\widehat{u}_1,\widehat{u}_2)=\max_{k,(u_1,u_2)}\mathcal{F}^k(u_1,u_2)$

The algorithm reports that the identity of the person in image $\mathcal{I}$ is $\widehat{k}$ and the face is centered at $(\widehat{u}_1,\widehat{u}_2)$.

3. The Face Identification System

The face identification system consists of three modules. The first is an off-line preprocessing module that designs the matching pursuit filters. The module designs two sets of filters: one for detecting and locating facial features and the other for identifying faces. This module also creates the initial gallery. The second module updates the gallery by adding or deleting individuals from the gallery. The third module is the on-line portion of the algorithm that takes as input images of unknown faces and returns their identity. This module consists of two stages. The first stage detects the face in the image and locates a small set of facial features. The locations of these facial features are fed to the second stage, which identifies the face. The identity is determined by comparison of the facial features of the unknown face with representations stored in the gallery. Figure 8 shows the system organization of the modules.

The face recognition algorithm is currently designed to identify people from full face frontal images using a small set of facial features (fig. 9). In the algorithm, a *feature* is a region of the image that contains a prominent facial feature. (For example, the left eye feature is the region of the face that contains the left eye.) The features are the tip and bridge of the nose, both eyes, and the interior of the face. The interior of the face feature is down-sampled by a factor of four.

Figure 8. System organization.

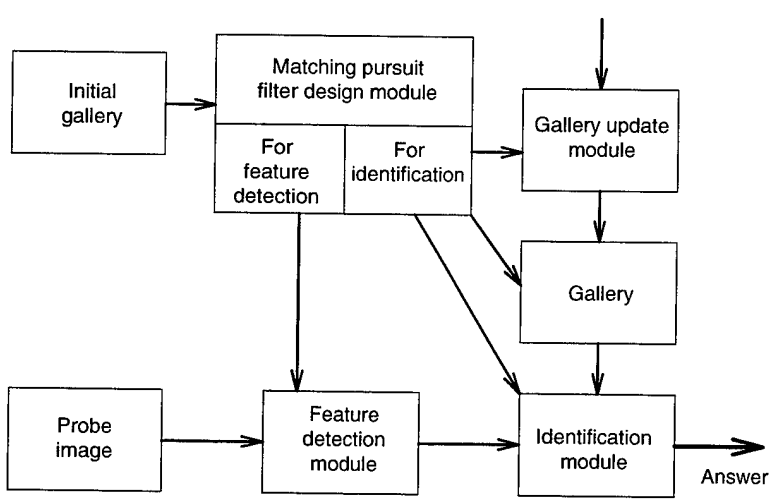
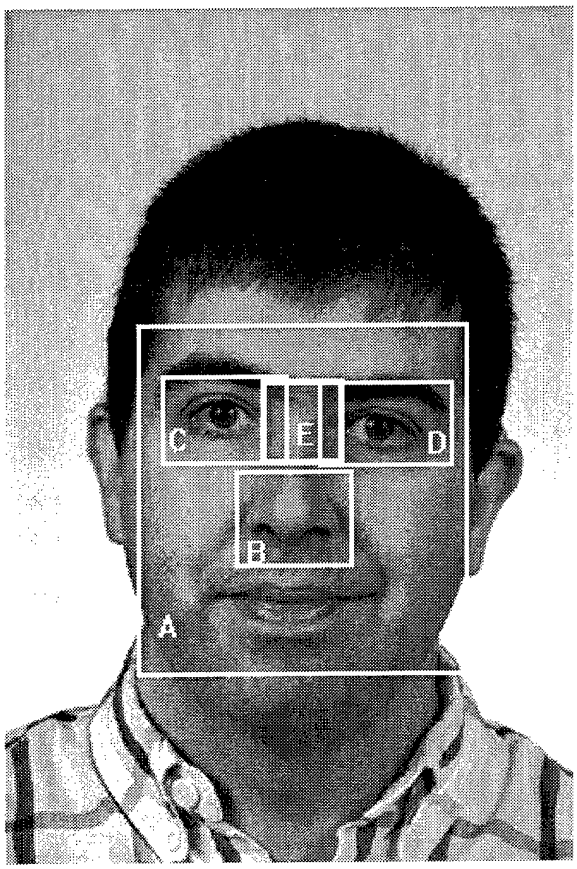


Figure 9. Facial features used: A is interior of face, B is tip of nose, C and D are left and right eyes, E is bridge of nose.



The nose and eyes were selected because they are the least variable facial features. Using these features increases the robustness of the algorithm with respect to variations in facial expression, facial hair, hair style, and the image background. The interior of the face feature encodes the overall shape and organization of the face.

The heart of the face identification system is the matching pursuit filters, which detect the face, locate facial features, and identify the face. The filters are used in both stages of the on-line module (the first stage detects and locates facial features, and the second stage identifies faces). The set of filters in the identification stage consists of five filters—one for each of the facial features (tip and bridge of the nose, left and right eyes, and interior of the face). There are four filters in the detection stage—tip of the nose, left and right eyes, and interior of the face. The location of the bridge of the nose is estimated as being mid-way between the eyes.

The preprocessing module designs both sets of matching pursuit filters. The training set for designing the filters is taken from the gallery. (For ex-

ample, the nose filters are trained from noses in images in the gallery.) The location of the feature in the gallery image is marked by a human operator. In the current implementation, only three points are marked, and accuracy is not critical. The points selected are the center of the eyes and the tip of the nose. The center of the eyes marks the location of the eye features; the tip of the nose marks the center of the interior of the face and the tip of the nose features. The center of the bridge of the nose is the average of the pixels marked as the center of the eyes.

In theory, the filters should be designed from all images in the gallery; however, for computational reasons this is not practical. In practice, filters are designed from a subset of the faces in the gallery. For the identification filters, the remaining faces are added to the gallery by the update procedure. The examples used to design the feature detection filters are limited to a subset of the images in the gallery, and the filters are not modified when the gallery is updated.

The gallery can be updated by either the deletion or addition of a person. The system adds a person to the gallery by computing the coefficient vector for each feature. The coefficient vector is computed by each feature being projected onto the appropriate wavelet expansion by equations (2) and (3). The new coefficient vectors are then added to the gallery. The locations of the features in new gallery images are marked by a human operator. The system deletes people from the gallery by removing their coefficient vectors from the gallery.

3.1 Feature Location

The feature location stage estimates the locations of the eyes, tip and bridge of nose, and center of the interior of the face. The first feature located is the interior of the face. In this stage, the module finds the face in the image by running the interior face filter on a decimated version of the probe image. The next set of features located is the eyes and the tip of the nose. The search for these facial features is guided by a priori knowledge of the geometry of the face and the estimated location of the center of the face: e.g., the right eye is to the right and above the center of the face. For locating these features, the full-size probe image is used. The estimated center of the bridge of the nose is midway between the estimated centers of the eyes. The locations are then passed to the identification stage. To avoid introducing a one-point failure in the algorithm, the feature-location stage passes multiple locations.

The first step in the feature location stage searches for the most likely locations of the face by running the interior face filter over the level 0 image (the

coarsest level image). The top N responses to the filter are reported as the most likely locations of the face. To avoid the situation where the hypothesized locations are neighbors, the algorithm uses the side condition that the locations must be a certain distance apart. Figure 10 contains pseudo-code for the face detection and feature location algorithm with $N = 1$. The value of N selected is a trade-off between the performance and speed of the algorithm. As N increases, the probability of correctly locating the face increases, as does the time to identify a face in a probe image. In the current implementation, $N = 3$.

The system uses the level 1 image to locate the remaining features by running the appropriate filter over a small region of the image. The feature is located at the maximum response of the filter in that region. The location and size of the region is based on the hypothesized location of the face, estimated error margins of the coarse-level processing, and the geometry of the face. For example, say the algorithm is searching for the left eye after the coarse level reports a detection. The coarse-level detector reports a pixel (u_1, u_2) that is the estimated location of the center of the face, which corresponds to the tip of the nose. From the gallery, it is known that the average translation from the tip of the nose to the center of the left eye is (t_1, t_2) and that a good error region is a p_1 by p_2 pixel box. If the error estimate for the coarse-level processing is an r_1 by r_2 pixel box, the left eye region is a $p_1 + r_1$ by $p_2 + r_2$ pixel box centered at $(u_1 + t_1, u_2 + t_2)$. The feature location algorithm reports the top N locations for each feature along with an error box for that feature.

Figure 10. Pseudo-code for face detection and feature location algorithm.

$\mathcal{I}_0$: level 0 image (decimated);
$\mathcal{I}_1$: level 1 image (original);
(t_1, t_2) : average translation between the nose and left eye;
$(-t_1, t_2)$: average translation between the nose and right eye;
$R : p_1 \times p_2$ box;

1: Search $\mathcal{I}_0$ image for the center of the face,
use face detection matching pursuit filter,
report that the center of face is pixel (u_1, u_2);
2: Search box R centered at (u_1, u_2) in $\mathcal{I}_1$ for the tip of the nose,
use the nose detection filter;
3: Search box R centered at $(u_1 + t_1, u_2 + t_2)$ in $\mathcal{I}_1$ for the left eye,
use the left eye detection filter;
4: Search box R centered at $(u_1 - t_1, u_2 + t_2)$ in $\mathcal{I}_1$ for the right eye,
use the right eye detection filter;
5: Report the bridge of nose as being mid-way between the eyes;

3.2 Identification of Individuals

The second stage of the on-line module identifies the face in the probe by comparing the coefficient vectors generated from the probe with the coefficient vectors stored in the gallery. Each face in the gallery is represented by five coefficient vectors, one for each feature. The system identifies a face in a probe by comparing coefficient vectors from the probe with the coefficient vectors of all the faces in the gallery. All faces in the gallery are compared with the probe, and a similarity score is produced for each comparison. The probe is identified as the face from the gallery with the highest similarity score.

The input to the identification stage is a set of hypothesized locations for each feature. The hypothesized location for feature j is region R_j, $1 \geq j \geq 5$ (one for each feature). If more than one set of feature locations is reported, then the algorithm is repeated for each of the feature sets, with the best match taken as the answer. The algorithm proceeds by computing the expansion for each pixel in region R_j; the resulting sets of coefficient vectors will be denoted by $\mathbf{a}^j(u_1, u_2)$.

The gallery consists of M individuals, and each individual is represented by N coefficient vectors. Let Λ^{ij} denote coefficient vector j of person i. The next step in the identification module is to find the best match between feature j of person i and the estimated locations of feature j in the probe image. This is done for each feature of each person. Let d_{ij} denote the score of the best match between feature j of person i and the region R_j. The score d_{ij} is computed as

$$d_{ij} = \max_{(u_1,u_2)\in R_j} d_\theta(\Lambda^{ij}, \mathbf{a}^j(u_1, u_2)).$$

For each person in the gallery, a total score is computed that represents the degree of similarity with the face in the probe image. The total score is a weighted sum of the best matches for the features of person i. Let

$$d_i = \sum_j w_j d_{ij}$$

be the score for the match between the unknown face and person i in the gallery, where w_j is the weight given to feature j. In this implementation, the weight for the interior of the face is 0.5, and the weights for the remaining features are 1.0. The unknown face is identified as the person with the maximum d_i. The system detects faces that are not in the gallery by setting a threshold δ. If the score of the best match is below δ, then the face is reported as not being in the gallery.

Figure 11 contains pseudo-code for the identification algorithm presented in this section. For clarity, the pseudo-code uses only two features, the face and the nose.

Figure 11. Pseudo-code for identification algorithm.

R_{face} : region hypothesized to contain the face;
R_{nose} : region hypothesized to contain the nose;
Λ^k_{face} : face coefficient vector for person k;
Λ^k_{nose} : nose coefficient vector for person k;

1: **do** $k = 1$ **to** k : number of images in database;
  2: Search R_{face} for maximum response to Λ^k_{face}; (figure 6)
    Let s^k_{face} be the maximum response;
  3: Search R_{nose} for maximum response to Λ^k_{nose};
    Let s^k_{nose} be the maximum response;
  4: Let $s_k = s^k_{\text{face}} + s^k_{\text{nose}}$,
    where s_k is the total score for person k;
5: **end do**;
6: Identify the unknown face as the person with the highest total score;

4. Experiments

To demonstrate the ability of this algorithm to recognize faces, I conducted four sets of experiments. The first decomposes and reconstructs a single nose; the second identifies faces from a large database; the third compares the performance of face recognition on infrared and visible imagery; and the fourth identifies faces from a database of mugshots from a law enforcement agency.

Matching pursuit filters were constructed from dictionaries of wavelets. The primary dictionary was composed of separable steerable filters [34], namely, the second partial derivatives of the Gaussian density and their Hilbert transforms. These filters were chosen because they are computationally less expensive than nonseparable filters. The primary dictionary was used in all four experiments. In the first experiment I also used a dictionary of Gabor wavelets (for computational ease, only the odd and even phase Gabor wavelets were used). Neither dictionary spanned the space of all possible images. Both dictionaries contained wavelets at four scales uniformly sampled in angular domain.

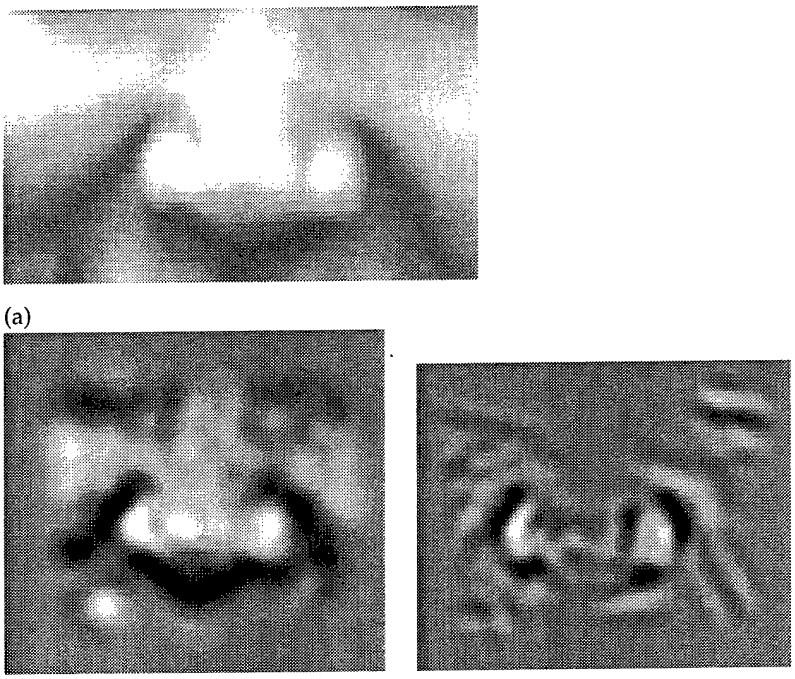
4.1 Decomposition of One Instance

The purpose of matching pursuit filters is to produce a representation that is tuned to a particular pattern recognition problem. However, if the training set consists of a single image, the filter design algorithm finds a compression-based representation. The first experiment demonstrated this by decomposing and reconstructing a single nose (fig. 12). Two different decompositions were computed from the two different dictionaries (the steerable filters and Gabor wavelets). Figure 12 shows the results of this experiment. Figure 12(a) is the original nose; figure 12(b) is the nose reconstructed from the first 75 terms of the separable steerable filter dictionary; and figure 12(c) is the nose reconstructed from the first 75 terms of a Gabor wavelet dictionary. This reduction of one example to a compression algorithm is what one would expect from the original work of Mallat and Zhang [11].

4.2 Face Recognition from FERET Images

The second experiment was the main experiment, where the face recognition algorithm was run on a gallery of 311 individuals with one image per

Figure 12. Decomposition of one image. (a) Original image. (b) Reconstruction using steerable filters. (c) Reconstruction using Gabor filters.



(a)

(b) (c)

person. The images were from the FERET database of facial images [35]. Figure 13 shows images from the FERET database used in the experiment. The **fa** images were in the gallery, and the **fb** images were probes. The training set for the identification filters consisted of images of 58 individuals, one image per person. Matching pursuit filters were constructed for each of the five facial features, and each filter had 30 coefficients (and is the same for all identification experiments). The algorithm computes the coefficient vectors for the remaining 253 people using the expansion from the identification filters. This procedure demonstrates that the filters generalize and that they do not need to be redesigned when a new person is added to the database. This is a critical concern for real-world applications.

The face recognition algorithm was run in two modes. The first mode tested the identification portion of the face recognition algorithm by providing the eye and nose coordinates to the algorithm. The second mode tested the complete face recognition algorithm by having the algorithm locate and identify the face. For both runs of the algorithm, the percentage of faces correctly identified in the top 1, 2, 3, and 4 matches is reported (table 1). The performance of the algorithm in both modes is virtually identical, indicating that in terms of system performance, the feature location algorithm is working nearly perfectly.

Figure 13. Face images from FERET database; **fa** images were placed in gallery and **fb** images were probes.

Table 1. Percentage of correctly identified faces for a database of 311 individuals.

	Correctly identified faces (%)	
Top match	Identification only	Location & identification
1	95.4	95.2
2	97.4	97.4
3	98.1	98.1
4	98.4	98.4

In the current implementation of the system, there are five features: the eyes, tip and bridge of the nose, and interior of the face. If a face is occluded or the hair style is changed, the number of usable features can decrease. To measure the effect of such changes on performance, I ran the algorithm using progressively fewer features. The first feature to be removed was the

interior of the face, because if one of the other features were corrupted, the interior of the face would also be corrupted. To isolate this effect from errors in locating facial features, I located the features in the probe set manually. Table 2 reports the results of this experiment and the combination of features tested.

4.3 Face Recognition from Infrared and Visible Imagery

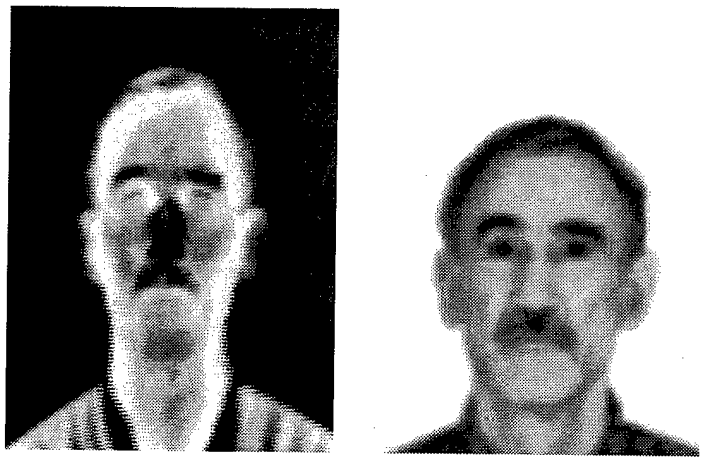
An area of growing interest is identifying faces from different modalities, in particular from infrared imagery. (For example, Wilder *et al.* [36] were among the first to address this issue in a comparison of the relative performance of three face recognition algorithms on infrared and visible images.) In this experiment, Wilder *et al.* measured recognition performance of three algorithms on both IR and visible images; these algorithms were matching pursuit filters, gray scale projection [37], and principal component analysis [14]. The goal of this study was not to compare performance between algorithms, but to measure their relative performance on the two modalities.

For the study, infrared and visible images of 101 subjects were collected. The infrared images were acquired with a Texas Instruments SMRTII uncooled sensor, which detects radiated heat. The visible images were collected with a CIDTEC 2250 CID camera. Figure 14 shows infrared and visible images of a subject. For an accurate measure of recognition performance, the images were scaled and rotated into a standard position. In both modalities, all algorithms were run on two sets of images. The results are reported in table 3. For each run, the table reports the percentage of correct answers in the top 1 and 2 matches.

Table 2. Performance as number of facial features is reduced.

	Correctly identified faces (%)				
Top match	All features	Eye and nose	Left eye & nose	Eye & bridge	
1	95.4	91.6	91.6	90.7	
2	97.4	93.4	94.5	92.3	
3	98.1	95.8	96.1	93.6	
4	98.4	96.8	96.8	93.9	
	Left & right eyes	Tip of nose & left eye	Left eye only	Nose only	Face only
1	87.1	89.4	81.4	80.1	78.5
2	88.7	91.3	85.5	84.6	82.0
3	88.7	93.2	86.8	85.9	85.2
4	88.7	93.4	88.4	87.1	87.5

Figure 14. Images of same person in (a) infrared and (b) visible.



(a) (b)

Table 3. Comparison of recognition of infrared and visible images in a database of 101 individuals.

	Correctly identified faces (%)					
	Gray scale projection		Eigenface		Matching pursuit filters	
Image set	Top match	Top 2 matches	Top match	Top 2 matches	Top match	Top 2 matches
Infrared set 1	91	94	86	92	89	94
Infrared set 2	90	93	86	89	94	96
Visible set 1	84	91	86	93	94	96
Visible set 2	83	90	89	94	96	98

If one has images of a person in two modalities, the simplest method of improving performance is to fuse the result from each modality. Each of the algorithms in this experiment produces a numeric score of the similarity between an unknown face and each image in the gallery. The identity is then determined by the best similarity score between the probe and gallery images. In this experiment, I fuse the two modalities by linear pooling of the similarity scores from the individual modalities. This was done for both image sets 1 and 2 (see table 3), with the results in table 4. In all cases, fusing the results improves performance to the saturation point. This suggests that a future area of research is studying methods of fusing infrared and visible imagery.

Table 4. Face recognition when data from infrared and visible images are fused (database of 101 individuals).

	Correctly identified faces (%)					
	Gray scale projection		Eigenface		Matching pursuit filters	
Image set	Top match	Top 2 matches	Top match	Top 2 matches	Top match	Top 2 matches
1	98	99	97	99	99	99
2	97	98	96	99	98	99

4.4 Mugshot Gallery

One potential application for a face identification algorithm is the electronic mugbook. In an electronic mugbook, the gallery consists of digital mugshots of known people, and a probe is a digital mugshot of an individual to be identified. In this experiment, I ran the algorithm mugshot data provided by a law enforcement agency.

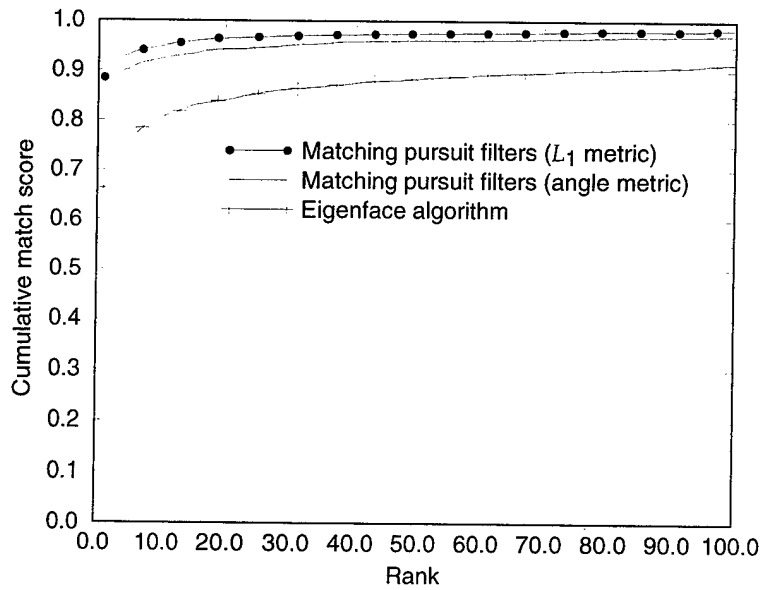
In the experiment, the gallery and probe sets were made up from digital mugshots of 2175 persons with two frontal images per person (side mugshots were not collected). The gallery was made up of one image from each pair, for 2175 persons; the probe set consisted of the other paired image, for 2126 individuals. The two images of each person were taken within a few minutes of each other; however, the subjects were not necessarily cooperative.

To obtain a performance baseline, I ran the dataset on an eigenface implementation. In this implementation, the images were placed in a standard position and masked; the pixel values inside the mask were then processed by a histogram equalization algorithm. The eigenfaces were trained on a subset of 500 images, and the faces were represented by the first 200 eigenvectors. For identification, the L_1 metric was used to measure the similarity between faces.

Two versions of the matching pursuit filter algorithm were run. In the first version, the similarity measure between coefficient vectors was the angle between vectors; in the second, the L_1 metric was the similarity measure. To assess the ability of matching pursuit filters to generalize across datasets, I used the expansions from the visible images in the IR/visible experiment (sect. 4.3).

The results are presented in figure 15 on a cumulative match plot. The x-axis is the rank of the ordering of the gallery from a match with a probe. The y-axis is the fraction of the probes correctly identified. The plot reports the fraction of the probes for which the correct answer is in the top

Figure 15. Performance of MPF and eigenfaces on mugshot gallery. Gallery size = 2175, probe set size = 2126.



n matches. For example, for the L_1 version of the matching pursuit algorithm, the correct match between a probe and gallery is in the top 10 for 0.95 of the probes. The results show that matching pursuit performs better than eigenfaces, and that the L_1 metric is better for matching pursuit than the angle between coefficient vectors. This experiment shows that matching pursuit filters can generalize across data sets and to larger datasets.

5. Conclusion

I describe a face identification algorithm that automatically locates facial features and identifies the located faces. The algorithm is based on a new class of filters called matching pursuit filters, which are wavelet expansions that are both data and problem adaptive. This means that matching pursuit filters are explicitly designed to solve the pattern recognition problems encountered in face recognition: locating faces and distinguishing between faces. Thus, two sets of filters were constructed: one set for the detection of facial features and one set for the identification of faces.

The algorithm was run on images from the FERET database and an associated database of visible and infrared images. The results of the visible and infrared study show that the performance of algorithms based on matching pursuit filters is comparable for both modalities. When the results of both modalities are fused, performance saturates (>97 percent correct identification). A larger database of infrared and visible images is required for an accurate assessment of the capabilities of multi-modal algorithms.

A number of algorithms in the literature report performance figures from tests run on the FERET database. Within this group, I restrict comparison to those algorithms that automatically process probe images. Each of these algorithms uses different training and tests, so the results quoted only provide an indirect comparison among the algorithms. The algorithm given here does better than the benchmark algorithm of Gutta *et al.* [28] (83 percent on a gallery of 200) and the correlation-based algorithm of Gordon [27] (72 percent on a gallery of 194). The performance of the algorithm described here is comparable to that of the eigenspace algorithm of Moghaddam and Pentland [30] (99.4 percent on a gallery of 150) and the dynamic link architecture algorithm of Wiskott *et al.* [22] (97.3 percent on a gallery of 300).

The eigenspace algorithm represents a face with 50 to 100 coefficients and represents the face in a global encoding, so that the algorithm is faster than it would be if local encoding were used. It is not, however, possible to explicitly account for local deformations. In contrast, dynamic link architecture explicitly handles local deformations with an elastic graph, but it requires ~4000 coefficients to represent a face. The method presented here is a compromise between these two methods. Each person is represented by 150 coefficients (30 coefficients for each of the five features).

In the present algorithm, the face is modeled as five geometric regions, which gives the algorithm the flexibility to adjust to deformations and variations in faces. I show this by measuring performance as the number of features is decreased. With all five features, the identification rate is 95.4 percent; with only one feature, it decreases to around 80 percent. The degradation in performance is graceful, with the greatest drop occurring when the number of features is reduced from two to one.

I have successfully demonstrated the performance of this algorithm in a number of experiments (on a gallery of 311, with infrared versus visible images, on a gallery of 2175 mugshots, and as the number of features decreases). This success shows that face identification algorithms based on matching pursuit filters are viable and can serve as a basis for a practical face identification system.

Acknowledgments

Portions of the work in this report are taken from my PhD dissertation. I thank the Operations Research program at RUTCOR, Rutgers University, for the program's support of a nontraditional operations research dissertation. Furthermore, I gratefully thank my thesis adviser, Yehuda Vardi, for his time, guidance, support, and many helpful conversations. Prof. Vardi's National Science Foundation Grant is acknowledged. Portions of this work were done under the FERET program, and the facial images are from the FERET database.

References

1. S. Haykin. *Neural Networks: A Comprehensive Foundation.* MacMillan, 1994.

2. T. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks* **5**:459–473, 1989.

3. T. E. Flick, L. K. Jones, R. G. Priest, and C. Herman. Pattern recognition using projection pursuit. *Pattern Recognition* **23**(12):1367–1376, 1990.

4. J.-N. Hwang, H. Li, M. Maechler, D. Martin, and J. Schimert. Projection pursuit learning networks for regression. *Eng. Applic. Artif. Intell.* **5**(3), 1992.

5. N. Intrator. Combining exploratory projection pursuit and projection pursuit regression with application to neural networks. *Neural Comput.* **5**:443–455, 1993.

6. J. Atick and A. N. Redlich. Convergent algorithm for sensory receptive field development. *Neural Comput.* **5**:45–60, 1993.

7. A. N. Redlich. Redundancy reduction as a strategy for unsupervised learning. *Neural Comput.* **5**:289–304, 1993.

8. M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Comput.* **42**:300–311, 1993.

9. J. G. Daugman. Complete discrete 2-D Gabor transform by neural networks for image analysis and compression. *IEEE Trans. Acoust. Speech Signal Process.* **36**(7):1169–1179, July 1988.

10. R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Trans. Info. Theory* **38**(2):713–718, 1992.

11. S. Mallat and Z. Zhang. Matching pursuit with time-frequency dictionaries. *IEEE Trans. Signal Process.* **41**(12):3397–3415, December 1993.

12. K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Trans. Image Process.* **2**(2):160–175, 1993.

13. K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, Orlando, FL, 1972.

14. M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neurosci.* **3**(1):71–86, 1991.

15. A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. Computer Vision and Pattern Recognition '94*, pp 84–91, 1994.

16. B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *Proc. Int. Conf. Computer Vision*, pp·786–793, 1995.

17. H. Murase and S. K. Nayar. Image spotting of 3D objects using the parametric eigenspace representation. In *Proc. 9th Scandinavian Conf. on Image Analysis*, pp 325–332, 1995.

18. H. Murase and S. K. Nayar. Visual learning and recognition of 3D objects from appearance. *Int. J. Comput. Vision* **14**(1):5–24, 1995.

19. P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs Fisherfaces: Recognition using class specific linear projection. In *Proc. 4th European Conf. Computer Vision*, pp 45–58, 1996.

20. K. Etemad and R. Chellappa. Discriminant analysis for recognition of human face images. *J. Opt. Soc. Am. A*, August 1997.

21. D. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. PAMI* **18**(8):831–836, 1996.

22. L. Wiskott, J.-M. Fellous, N. Kruger, and C. von der Malsburg. Face recognition and gender determination. In *Int. Workshop on Automatic Face and Gesture Recognition*, M. Bichsel, editor, pp 92–97, 1995.

23. R.P.N. Rao and D. H. Ballard. An active vision architecture based on iconic representation. *Artif. Intell.* **78**:461–505, 1995.

24. R.P.N. Rao and D. H. Ballard. Object indexing using an iconic sparse distributed memory. In *Proc. Int. Conf. Computer Vision*, pp 24–31, 1995.

25. P. Viola. Feature-based recognition of objects. In *Proc. AAAI Fall Symposium on Learning and Computer Vision*, pp 60–64, 1993.

26. P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face-recognition algorithms. *Image Vision Comput. J.*, to appear, 1997.

27. G. G. Gordon. Face recognition from frontal and profile views. In *Int. Workshop on Automatic Face and Gesture Recognition*, M. Bichsel, editor, pp 47–52, 1995.

28. S. Gutta, J. Huang, D. Singh, I. Shah, B. Takacs, and H. Wechsler. Benchmark studies on face recognition. In *Int. Workshop on Automatic Face and Gesture Recognition*, M. Bichsel, editor, pp 227–231, 1995.

29. T. Maurer and C. von der Malsburg. Single-view based recognition of faces rotated in depth. In *Int. Workshop on Automatic Face and Gesture Recognition*, M. Bichsel, editor, pp 248–253, 1995.

30. B. Moghaddam and A. Pentland. Face recognition using view-based and modular eigenspaces. In *Proc. SPIE Conf. Automatic Systems for the Identification and Inspection of Humans*, SPIE **2277**, pp 12–21, 1994.

31. D. Swets and J. Weng. Discriminant analysis and eigenspace partition tree for face and object recognition from views. In *2nd Int. Conf. Automatic Face and Gesture Recognition*, pp 192–197, 1996.

32. I. Cox, J. Ghosen, and P. Yianilos. Feature-based face recognition using mixture-distance. In *Proc. Computer Vision and Pattern Recognition '96*, pp 209–216, 1996.

33. M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding*. Prentice Hall PTR, 1995.

34. W. T. Freeman and E. H. Adelson. The design of steerable filters. *IEEE Trans. PAMI* **13**(9):891–906, 1991.

35. P. Rauss, P. J. Phillips, A. T. DePersia, and M. Hamilton. Face recognition technology program overview and results. In *25th AIPR Workshop: Emerging Applications of Computer Vision*, SPIE **2962**, pp 253–263, 1996.

36. J. Wilder, P. J. Phillips, C. Jiang, and S. Wiener. Comparison of visible and infrared imagery for face recognition. In *2nd Int. Conf. Automatic Face and Gesture Recognition*, pp 182–187, 1996.

37. J. Wilder. Face recognition using transform coding of gray scale projections and the neural tree network. In *Artificial Neural Networks with Applications in Speech and Vision*, R. J. Mammone, editor, pp 520–536, Chapman Hall, 1994.

Distribution

Admnstr
Defns Techl Info Ctr
Attn DTIC-OCP
8725 John J Kingman Rd Ste 0944
FT Belvoir VA 22060-6218

Ofc of the Secy of Defns
Attn ODDRE (R&AT) S Gontarek
The Pentagon
Washington DC 20301-3080

US Army Train & Doctrine Cmd
Battle Lab Integration & Techl Dirctrt
Attn ATCD-B J A Klevecz
FT Monroe VA 23651-5850

CECOM
Attn PM GPS COL S Young
FT Monmouth NJ 07703

DARPA
Attn B Kaspar
3701 N Fairfax Dr
Arlington VA 22203-1714

Dept of the Army (OASA) RDA
Attn SARD-PT R Saunders
103 Army
Washington DC 20301-0103

Dir of Assessment and Eval
Attn SARD-ZD H K Fallin Jr
103 Army Pentagon Rm 2E673
Washington DC 20310-0163

Hdqtrs Dept of the Army
Attn DAMO-FDQ D Schmidt
400 Army Pentagon
Washington DC 20310-0460

MICOM RDEC
Attn AMSMI-RD W C McCorkle
Redstone Arsenal AL 35898-5240

OSD
Attn OUSD(A&T)/ODDDR&E(R) R J Trew
Attn OUSD(A&T)/ODDDR&E(R) S Ahmadi
The Pentagon
Washington DC 20301-7100

US Army Edgewood Rsrch, Dev, & Engrg Ctr
Attn SCBRD-TD J Vervier
Aberdeen Proving Ground MD 21010-5423

US Army Info Sys Engrg Cmd
Attn ASQB-OTD F Jenia
FT Huachuca AZ 85613-5300

US Army Materiel Sys Analysis Agency
Attn AMXSY-D J McCarthy
Aberdeen Proving Ground MD 21005-5071

US Army Natick Rsrch, Dev, & Engrg Ctr
Acting Techl Dir
Attn SSCNC-T P Brandler
Natick MA 01760-5002

US Army Rsrch Ofc
4300 S Miami Blvd
Research Triangle Park NC 27709

US Army Simulation, Train, & Instrmntn Cmd
Attn J Stahl
12350 Research Parkway
Orlando FL 32826-3726

US Army Tank-Automotive Cmd Rsrch, Dev, & Engrg Ctr
Attn AMSTA-TA J Chapin
Warren MI 48397-5000

US Army Tank-Automtv & Armaments Cmd
Attn AMSTA-AR-TD C Spinelli
Bldg 1
Picatinny Arsenal NJ 07806-5000

US Army Test & Eval Cmd
Attn R G Pollard III
Aberdeen Proving Ground MD 21005-5055

US Military Academy
Dept of Mathematical Sci
Attn MDN-A MAJ MD Phillips
West Point NY 10996

Ofc of the Dir Rsrch and Engrg
Attn R Menz
Pentagon Rm 3E1089
Washington DC 20301-3080

Distribution (cont'd)

ARL Electromag Group
Attn Campus Mail Code F0250 A Tucker
University of Texas
Austin TX 78712

MA Instit of Techlgy
Brain Sci Dept & A. I. Lab
Attn E25-218 T Poggio
45 Carleton Stret
Cambridge MA 02142

MIT Dept Program Media Arts & Sci
Attn A Pentland
77 Massachusetts Ave
Cambridge MA 2139

Univ of Maryland Dept of Elec Engrg
Attn R Chellappa
Rm 2365 A V Williams Bldg
College Park MD 20742-3285

Dir for MANPRINT
Ofc of the Deputy Chief of Staff for Personnel
Attn J Hiller
The Pentagon Rm 2C733
Washington DC 20310-0300

ERIM
Attn C Dwan
Attn J Ackenhusen
1975 Green Rd
Ann Arbor MI 48105

Hicks & Associates, Inc
Attn G. Singley III
1710 Goodrich Dr. Ste 1300
McLean VA 22102

Natl Institute of Justice Techlgy Dev
Attn A T DePersia
633 Indiana Ave NW
Washington DC 20531

National Institute for Standards & Technology
Attn J Phillips
Bldg 225 Rm A216
Gaithersburg, MD 20899

SAIC
Attn F Shields
4001 N Fairfax Dr Ste 300
Arlington VA 22203

US Army Rsrch Lab
Attn AMSRL-CI-LL Tech Lib (3 copies)
Attn AMSRL-D J Lyons
Attn AMSRL-DD J Rocchio
Attn AMSRL-CS-AL-TA Mail & Records Mgmt
Attn AMSRL-CS-AL-TP Techl Pub (3 copies)
Attn AMSRL-IS P Emmerman
Attn AMSRL-IS-CI P David
Attn AMSRL-IS-CS J D Gantt
Attn AMSRL-SE J M Miller
Attn AMSRL-SE-E J Pellegrino
Attn AMSRL-SE-SE D Nguyen
Attn AMSRL-SE-SE L Bennett
Attn AMSRL-SE-SE M Vrabel
Attn AMSRL-SE-SE N Nasrabadi
Attn AMSRL-SE-SE P Rauss
Attn AMSRL-SE-SE S Der
Attn AMSRL-SE-SI T Kipp
Adelphi MD 20783-1197

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
October 1998

3. REPORT TYPE AND DATES COVERED
Interim, 6/1/97 to 6/30/97

4. TITLE AND SUBTITLE Matching Pursuit Filters Applied to Face Identification

5. FUNDING NUMBERS
DA PR: —
PE: 62120A

6. AUTHOR(S) P. Jonathon Phillips (now with the National Institute of Standards and Technology, jonathon@nist.gov)
ARL POC: Matthew Thielke

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Attn: AMSRL-SE-SE email: mthielke@arl.mil
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-1487

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
DoD Counterdrug Technology Development Program Office
Naval Surface Warfare Center, Dahlgren Division B07
17320 Dahlgren Road
Dahlgren, VA 22448-5100

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
ARL PR: 7NMAMM
AMS code: 622120.H16

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A face identification algorithm is presented that automatically processes an unknown image by locating and identifying the face. The heart of the algorithm is the use of pursuit filters. A matching pursuit filter is an adapted wavelet expansion, where the expansion is adapted to both the data and the pattern recognition problem being addressed. For identification, the filters find the features that differentiate among faces, whereas for detection, the filters encode the similarities among faces. The filters are designed through a simultaneous decomposition of a training set into a two-dimensional wavelet expansion. This yields a representation that is explicitly two-dimensional and encodes information locally.

The algorithm uses coarse to fine processing to locate a small set of key facial features, which are restricted to the nose and eye regions of the face. The result is an algorithm that is robust to variations infacial expression, hair style, and the surrounding environment. Based on the locations of the facial features, the identification module searches the database for the identity of the unknown face using matching pursuit filters to make the identification.

The algorithm was demonstrated on three sets of images. The first set was images from the FERET database. The second set was infrared and visible images of the same people. (These two sets allowed the examination of algorithm performance on infrared and visible images individually, and on fused data from both modalities.) The third set of images was mugshot data from a law enforcement application.

14. SUBJECT TERMS
Face recognition, FERET, wavelets, adaptive expansion

15. NUMBER OF PAGES
42

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102