MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1591 November, 1996

Complex Feature Recognition: A Bayesian Approach for Learning to Recognize Objects

Paul A. Viola

This publication can be retrieved by anonymous ftp to publications.ai.mit.edu.

Abstract

We have developed a new Bayesian framework for visual object recognition which is based on the insight that images of objects can be modeled as a conjunction of local features. This framework can be used to both derive an object recognition algorithm and an algorithm for learning the features themselves. The overall approach, called complex feature recognition or CFR, is unique for several reasons: it is broadly applicable to a wide range of object types, it makes constructing object models easy, it is capable of identifying either the class or the identity of an object, and it is computationally efficient – requiring time proportional to the size of the image.

Instead of a single simple feature such as an edge, CFR uses a large set of complex features that are learned from experience with model objects. The response of a single complex feature contains much more class information than does a single edge. This significantly reduces the number of possible correspondences between the model and the image. In addition, CFR takes advantage of a type of image processing called *oriented energy*. Oriented energy is used to efficiently pre-process the image to eliminate some of the difficulties associated with changes in lighting and pose.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for this research was provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-96-1-0311.

1 Introduction

The essential problem of object recognition is this: given an image, what known object is most likely to have generated it? There are a huge variety of approaches to this problem. From these we can extract at least one critical insight. There is no simple relationship between the image and the objects present. Among the confounding influences are pose, lighting, clutter and occlusion. As a result, many researchers have eschewed the use of the image itself as the representation for recognition. Instead they choose to define and identify simple image features that are supposed to capture the important characteristics of the image (Ballard, 1981), (Bolles and Cain, 1982), (Grimson and Lozano-Perez, 1984). A typical example of such a feature is an intensity edge. There are three main motivations for using simple features. First, it is assumed that simple features are detectable under a wide variety of pose and lighting changes. Second, the resulting image representation is compact and discrete, consisting of a list of features and their positions. Third, it is hypothesized that the position of these features in a novel image of an object can be predicted from knowledge of their positions in other images. In many ways these motivations are justified. But there is one main difficulty associated with using simple features for recognition. It is very difficult to determine which feature of an image corresponds to each feature in an object – the correspondence problem. The feature itself, since it is simple, does not provide any constraint on the match.

We propose a novel approach to image representation that does not use a single predefined feature. Instead, we use a large set of complex features that are *learned* from experience with model objects. The response of a single complex feature contains much more class information than does a single edge. This significantly reduces the number of possible correspondences between the model and the image.

In order to better understand and more clearly derive the results in this paper, a probabilistic framework for the formation of images is defined. This framework can be used to predict what an image of a particular object looks like. From this framework, Bayes' theorem can be used to derive the CFR recognition algorithm. We believe that this formal approach makes the assumption underlying CFR, and related techniques, clear. Sections 2 and 3. describe the Bayesian framework and CFR respectively.

The performance of the CFR recognition procedure critically depends on having an appropriate set of complex features. Without good features the generative process will fail to accurately capture the appearance of an object and the recognition performance of CFR will rapidly degrade. An additional side-benefit of the formal framework used to present CFR is that it can be used to derive a principled mechanism for learning appropriate features. Though this is perhaps the most novel aspect of this research, the learning rule for features can only be derived once the description of the CFR framework is complete. Discussion of this learning rule is in Section 4.

In order to improve the generalization of CFR to novel poses and different illumination, images are processed to extract information about rapid changes in intensity.

Similar pre-processing can be found in the visual cortex of primates (see (Kandel and Schwartz, 1985) for example) and underlies the computational definition of the intensity edge by (Marr and Hildreth, 1980). Rather than the discrete detection of intensity edges, CFR instead uses a continuous measure of the "edge-ness" of pixels. The "edge-ness" of a pixel is proportional to the energy in a number of oriented band-pass filters centered on the pixel. This representation and its advantages are described in Section 5.

In Section 6 a number of CFR experiments are described. In these experiments CFR is shown to work both with human faces and real objects. Finally a number of extensions to the CFR framework are proposed.

2 A Generative Process for Images

A generative process is much like a computer graphics rendering system. A rendering system takes an object description, information about illumination and pose and it generates a life-like image of the object. One naive procedure for recognizing a novel image is to generate all possible images that might result from a model object. If one of these synthetic images matches the novel image "well" then there is good evidence that the novel image is an example of that model.

While computer graphics is a deterministic process (i.e. for every object and pose there is a single unique image) the world is more unpredictable. Some noise or unmodelled variable may have changed the rendered image before it is recorded by a camera. To address this lack of predictability, a probabilistic generative process defines a probability density over the space of possible images.

More formally, given an image I, an object model M, and a pose β a generative process allows us to compute:

$$P(I \mid M\beta) \tag{1}$$

the probability of an image given that we know which object is present and its pose. Bayes' theorem then tells us:

$$P(M,\beta \mid I) = \frac{P(I \mid M,\beta)P(M,\beta)}{P(I)} \tag{2}$$

We can now define an object recognition algorithm that returns the object that is the most probable[1]:

$$\operatorname*{argmax}_{M} \sum_{\beta} P(M,\beta \mid I). \tag{3}$$

This type of approach is not new. Most well-known object recognition systems can be formulated as a search for the model that makes the image most likely (see

[1]Since we do not know what the pose of the object is we choose to "integrate out" the unknown variable. Alternatively we could find β that makes O most likely. We explore this option later in the paper.

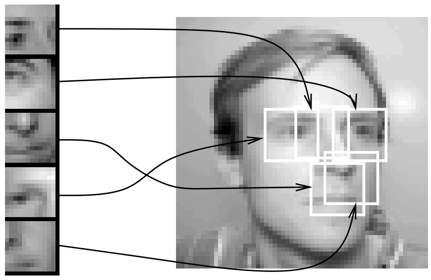

Figure 1: A schematic depiction of an image, a set of complex features, and a partial representation of that image given these features. The arrows between the five different features on the left and the five white boxes that lie over the image describe the positions of the features that best represent the image.

for example (Wells III, 1991) which makes this analogy very explicit). Of course the details of these algorithms can be quite different. Some algorithms use the input image directly, comparing the input image and the predicted image directly. Most techniques that use correlation for image matching fall into this category. Other algorithms assume that images are well described by the positions of simple image features, like edges. The image features are then compared to the predicted features from a generative process.

Our generative process is really somewhere between the direct and feature based approaches. Like feature based approaches, it uses features to represent images. But, rather than extracting and localizing a single type of simple feature, a more complex yet still local set of features is defined. Like direct techniques, it makes detailed predictions about the intensity of pixels in the image.

To emphasize that the features used in our system are more complex than is typical we call it Complex Feature Recognition (CFR). In CFR every image is a collection of distinct complex features (see figure 1). Complex features are chosen so that they are *distinct* and *stable*. A distinct feature is one that appears no more than a few times in any image and is correlated with a particular object or class of objects. Simple features, especially edges, are decidedly not distinct. Stability has two related meanings: i) the position of a stable feature changes slowly as the pose of an object changes slowly; ii) a stable feature is present in a range of views of an object about some canonical view. Simple features are intended to be, though they often aren't, stable. In our current implementation, and in our analysis, complex features resemble templates, but our formulation is general enough to admit a number of different feature representation mechanisms.

2.1 Some Examples

A few simple examples will help to motivate the thinking behind CFR. Clearly a picture of a person, if suitably normalized to remove some of the dependency on

Figure 2: A typical set of images of a single person. In this case there are fifteen views that are centered around the "direct forward" view. All of the other face images used in this paper take this form. The face data used in this paper came from David Beymer of the MIT AI Laboratory.

lighting, might be an excellent complex feature. It is distinct; there aren't many objects that look like a person that aren't. But, a picture is not a stable complex feature. Intuition tells us that a small change in object pose would rapidly make the picture a poor predictor of the image. For instance in figure 2 we see images in 15 canonical poses of a person (these poses are numbered left to right and top to bottom starting from 0). As we can see, and we will soon quantify, appearance of these images changes quite rapidly as pose varies.

While a picture of the entire object may be a poor complex feature, are there other more local pictures that would work better? Figure 3 addresses this question empirically. On the left of the figure, labeled (a), are representative images of three different people. On the top right, labeled (b) is a candidate local feature, and a graph representing its response to the 15 canonical poses of these three people. The feature is selected from pose number 2 of the first person. While chosen fairly arbitrarily, it is roughly the largest possible square sub-region of the the image that does not contain a lot of background or any non-face regions. The graph plots for each of these three people, pose number versus a measure of "nearness" between the complex feature and the image[2]. Note that since the feature is taken from pose 2, the feature is nearest to pose 2 of the first person. Unfortunately, this feature does not act to distinguish the three people. No simple threshold on feature response would suffice to identify person 1.

This sort of feature is not entirely useless. It may be useful for identifying images of person 1 in a limited number of poses – the ones near pose 2, 7 or 12 (notice that these poses are actually very similar). We have shown a threshold for this type of

[2]We use the maximal value of the normalized correlation between the feature and the image as a measure of image distance. Normalized correlation is a widely used matching metric that eliminates some of the dependency on lighting (Brunelli and Poggio, 1992). More on this later.

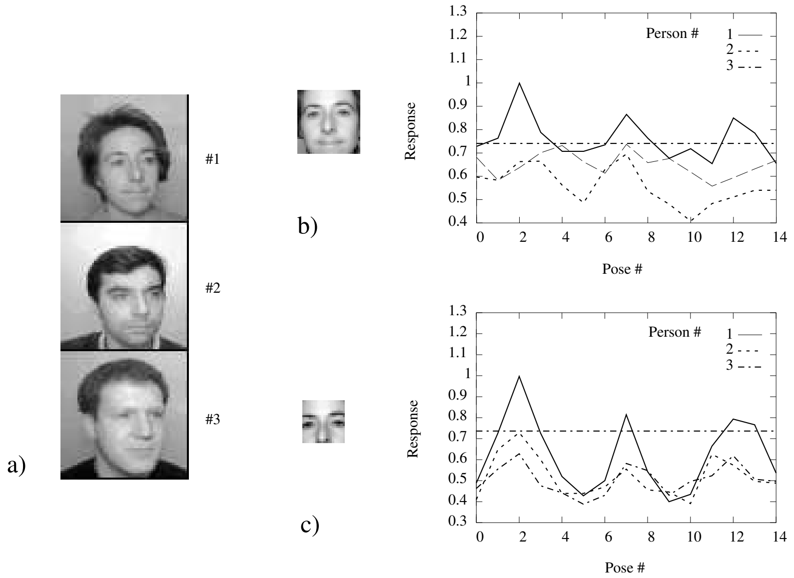

Figure 3: The responses of simple grey level features. See text for a more complete description. (a) example images of three people. (b) a feature from the pose 2 of person 1 and its response versus fifteen views of the three people (c) a slightly smaller feature taken from the same person, and its responses.

discrimination as a line at about 0.75. Labeled (c) is a another feature and a similar graph. This feature has been selected arbitrarily, but this time to represent a smaller more localized part of the face. Once again we can see that the feature, by itself, is not very good for recognition.

Is it possible to build complex features that are distinct and stable? With some extra machinery, the answer is a qualified yes. In figure 4 we show a graph of the output of a arguably much better feature. We see that for a large number of poses this feature acts to discriminate person 1 from 2 and 3. Furthermore, it does this by a much larger margin. Is the position where the feature appears stable across pose changes? Figure 5 contains 5 representative images. We have labeled the location where the feature responds most strongly with a white square. While the feature is clearly not responding to some true 3D location on the face, it seems to respond to the local region of the forehead. This is not the universal behavior of learned features. Other features often do a much better job of localization at the expense of generalization across views.

CFR's complex features differ from the simple image templates shown in Figure 3 in two major ways. First, complex features are not matched directly to the pixels of the image. Instead we match an easily computed intermediate representation called *oriented energy*. An oriented energy representation of an image is in fact several images, one for each of a number of orientations. The value of a particular pixel in the vertical energy image, is related to the likelihood that there is a vertical edge near that

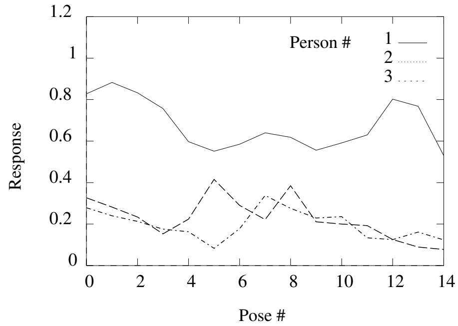

Figure 4: A graph of the performance of a learned complex feature on the same data as the graphs above.

Figure 5: The white square labels the location where a forehead feature detector responds most strongly.

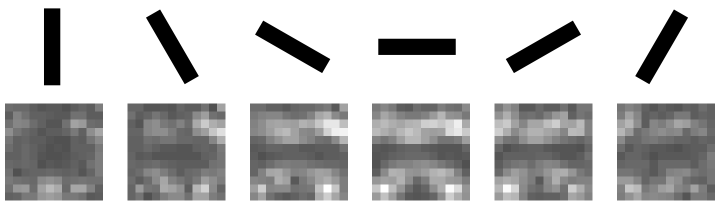

Figure 6: A raw untrained oriented energy feature that has been selected from the person 2, in pose 2. The feature represents the area near the forehead. The oriented energy feature is made up of 6 images one for each of six orientations. The first measures vertical energy, the fourth horizontal energy. The remaining orientations are evenly distributed. Notice that the hairline and eyebrows in the third or horizontal image are accentuated.

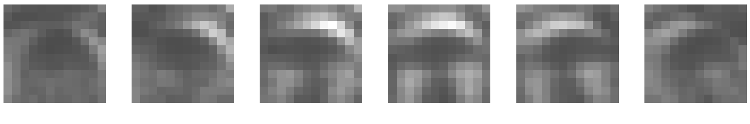

Figure 7: A feature trained to respond strongly to person number 1. Starting from the one shown in the previous figure, which was taken from person number 2, CFR has learned salient properties of person number 1.

pixel in the original image. Figure 6 shows a typical oriented energy feature. In this paper six orientations are used, though this is not critical. The feature representation is discussed in more detail in Section 5.

The second difference between an image template and a complex feature is that it result of a feature learning procedure. The complex feature has been adjusted so that it responds strongly to all of the example images of person number 1. The learning process allows CFR to discover features which are effective for classifying an object across a wide variety of poses. Figure 7 shows a feature which is the result of tuning the feature in Figure 6 to more closely model person 1. The details of the feature learning algorithm are described in Section 4.

Clearly one cannot attempt to build a recognition system around a single feature, even a very good one. CFR uses many features trained on a wide variety of objects and poses. The types of features that CFR uses have a very different flavor from those used by simple recognition systems. Each feature is correlated, though not exclusively, with particular objects or classes of objects. Each feature is detectable from a set of poses about some nominal pose. Finally, each feature is localizable across these poses. This allows us to use the relative positions of the features as additional information for recognition.

In the next section a theory for complex feature recognition is outlined. This theory provides a means for analyzing and understanding the computations that are used to represent and recognize images.

3 The Theory of Complex Features

Let I be a random variable from which images are drawn. An image is a vector of pixel values which have a bounded range of R. The pixels in these images need not be intensities measured with a camera. They may be any pixelated representation of an image. If multiple pixelated representation are available, as is the case for oriented energy, each pixel can be viewed as having a vector value, with one dimension for each pixelated representation. Since the use of an explicit vector notation for the pixel values leads to additional notational complexity, derivations will use a scalar notation for pixels. It is not difficult to rederive this theory for vector valued pixels.

We are also given a set of complex features $\{f_i\}$, such that n_i is the number of pixels in f_i. Let each object be represented a collection of models $\{M\}$. Each object may require several models in order to capture the variation in appearance due to changes in pose. Each model is drawn from a pair of random variables D and L. D is an indicator vector and L is a vector of locations. When a feature f_i is present in an image, $D_i = 1$, otherwise it is 0. When $D_i = 1$, L_i is the location of f_i in the image. Let $S()$ be a sub-window function on images such that $S(I, L_i)$ is a sub-window of I that lies at position L_i (see figure 8). We can now define the conditional probability of a particular image sub-window:

$$P(S(I, l_i) \mid D_i = 1, L_i = l_i, f_i) = N(S(I, l_i); f_i, \Sigma) \tag{4}$$

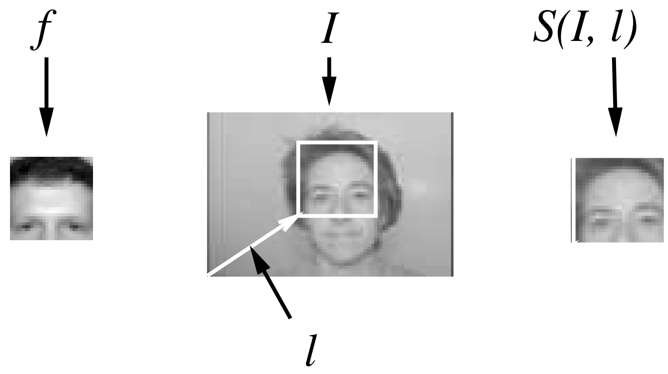

Figure 8: A diagrammatic depiction of the operation of the sub-window function S. Given a feature f, an image I and a location l, $S(I,l)$ returns a sub-window of I that that lies of at location l and is of the same size as f.

and

$$P(S(I,l_i) \mid D_i = 0, L_i = l_i, f_i) = + \left(\frac{1}{R}\right)^{|f|} \tag{5}$$

where $N()$ is the normal distribution over the the pixels of $S(I,l_i)$ with mean f and covariance matrix Σ. These equations can be interpreted in the following way: if $D_i = 1$, then the probability of an image is a function of the distance between the pixels of $S(I,l_i)$ and f. Otherwise, we assume that each of the pixels in $S(I,l_i)$ are uniformly distributed (each pixel's density being $\frac{1}{R}$).

To reiterate, the variables I, D and L are considered to be random variables. Since D and L are vectors, their components, L_i and D_i are also random variables. Events drawn from these distributions will be denoted with small letters, especially d, l, d_i and l_i. In some derivations to simplify notation $P(D = d)$ and $P(d)$ will be used to denote the same thing, namely the probability that the random variable D will take on the value d.

If we assume that the features never overlap and they are independent then the probability density of an image given $M = (d, l)$ is[3]:

$$P(I \mid d, l) = \prod_i P(S(I,l_i) \mid d_i, l_i, f_i) \frac{1}{R^u} \tag{6}$$

where u is the number of pixels in the image that remain unexplained by any feature.

Following Bayes' theorem we can now compute the probability of a model given

[3]For clarity we will use a derivation that assumes that the features never overlap and that they independent. An alternative formulation exists in which dependent, overlapping features can be used. Many of the computation that are tractable in the independent formulation, become significantly less tractable in the dependent formulation.

an image:

$$\begin{aligned} P(d, l \mid I) &= \frac{P(I \mid d, l) P(d, l)}{P(I)} && (7) \\ P(d \mid I) &= \sum_l \prod_i \frac{P(I \mid d_i, l_i) P(d_i, l_i)}{P(I)} && (8) \\ &= \prod_i \sum_{l_i} \frac{P(I \mid d_i, l_i) P(d_i, l_i)}{P(I)} && (9) \\ &\approx \prod_i \max_{l_i} \frac{P(I \mid d_i, l_i) P(d_i, l_i)}{P(I)} && (10) \end{aligned}$$

Note, in Equation 9 the sum over vectors of locations has be split into separate sums over each feature location and moved inside the product. This can be done because the features are assumed independent.

Equation 10 can be used to define an algorithm for recognizing which of a set of objects appears in an image. Given a collection of models, $\{M = (d^M, l^M)\}$, find the model that is most likely. Of course a significant difficulty remains, that of finding and computing the object models. A straightforward scheme for building a model is to obtain a segmented image of an object, and pick d_i and l_i to be the most likely given the image:

$$(\hat{d}_i, \hat{l}_i) = \operatorname*{argmax}_{d_i, l_i} P(D_i = d_i, L_i = l_i \mid I). \qquad (11)$$

While this has a pleasing simplicity we must be wary of the case when $P(D_i = 1 \mid I)$ is not significantly greater than $P(D_i = 0 \mid I)$. This is true when the presence or absence of f_i is ambiguous. Picking a single value for $\hat{d}_i$ in this case is misleading. The real situation is that $\hat{d}_i$ is about equally likely to be 1 or 0. Worse it confuses the two very distinct types of models: $P(D_i = 1 \mid I) \gg P(D_i = 0 \mid I)$ and $P(D_i = 1 \mid I) = P(D_i = 0 \mid I) + \epsilon$. In experiments this type of maximum a posteriori model does not work well.

An alternative type of object model retains explicit information about $P(D_i \mid I)$:

$$\hat{d}_i = \max_{l_i} P(D_i = d_i, L_i = l_i \mid I) \qquad (12)$$

and

$$\hat{l}_i = \operatorname*{argmax}_{l_i} P(D_i = d_i, L_i = l_i \mid I). \qquad (13)$$

Note that $\hat{d}$, a vector of numbers between zero and one, is not an event of D, which is a binary vector. $\hat{d}$ is no longer the most likely value for d_i, instead it is an estimate for the distribution of D_i. The resulting object models, $\{M = (\hat{d}^M, \hat{l}^M)\}$ are probabilistic. The probability of an image given such a model is now really a mixture distribution:

$$P(I \mid M) = \sum_{d,l} P(I \mid d, l) P(d, l \mid M). \qquad (14)$$

There are two distinct ways to define a recognition algorithm for a probabilistic model. We could simply use Bayes' theorem once again:

$$\operatorname*{argmin}_M P(M \mid I) = \frac{P(I \mid M)P(M)}{P(I)}. \tag{15}$$

Alternatively we can find the model whose probability density over the feature indicator variables is most closely matched by the image:

$$\operatorname*{argmin}_m G(P(D \mid M), P(D \mid I)), \tag{16}$$

where $G(p, q)$ is a density function distance measure: it returns 0 if p and q are equal, and larger values as p and q diverge. By defining

$$\hat{d}^I_j = P(D_j \mid M), \tag{17}$$

G becomes a measure which compares the vectors $\hat{d}^I$ and $\hat{d}^M$. There are a number of reasonable candidates for G, perhaps the best motivated is the cross entropy or asymmetric divergence (see (Cover and Thomas, 1991) for an excellent review entropy and divergence). For simplicity, we have chosen to use the squared difference,

$$G(\hat{d}^I, \hat{d}^M) = \left|\hat{d}^I - \hat{d}^M\right|^2. \tag{18}$$

The resulting recognition algorithm is called *CFR-MEM*, because it explicitly memorizes the distribution of features in each of the model images.

We have also explored another scheme for classifying images. Experiments have shown us that a large number of object models are often necessary to correctly classify novel object images. While there currently is no formal analysis of this problem, one manifest issue is that not all features are correlated with object identity. These distractor features vary widely over similar views of the same object. Since each feature is treated uniformly by the distance function G, the distractor features corrupt an otherwise good fit between model and image. One could attempt to generalize G so that it weights "good" features more than distracting features. But one object's good feature is often another object's distractor. This would force us to find a different G for each object. A more direct approach is to learn a classifier. A classifier is a function $C(\vec{v})$ that computes object identity. We have chosen to use a multi-layer perceptron, also known as a neural network, to learn a classifier (Rumelhart, Hinton and Williams, 1986). Briefly, a neural network is a clever way of parameterizing a function $C(\vec{v}, W)$ with a set of weights W. The weights are then learned by defining a training set of pairs $\{\vec{v}_j, \vec{O}_j\}$, that label each input vector with its corresponding object. The $\vec{O}_j$'s are vectors where the i'th component is 1 if the i'th object is present and 0 otherwise. A set of weights are selected that minimize the error over the training set, $E = \sum_j |\vec{O}_j - C(\vec{v}_j, W)|^2$, by using a form of gradient descent. We call this recognition algorithm *CFR-DISC* (DISC for discriminator). Interestingly, we need not train the network to compute the identity of the object. Object class, like "face" versus "car", is an equally well defined target.

In this section we have derived two algorithms for recognizing objects with a set of complex features. These derivations have assumed that some set of complex features is available. In practice however, these algorithms depend on the selection of appropriate features. In the next section we will derive an algorithm for learning a set of features that fit images well.

4 Learning Features

The Bayesian approach to detection, or recognition, critically depends on the effectiveness of the generative process. In other words, for each of a set of training images there should be at least one likely CFR model. If it is impossible to model the training images of an object, then it will be difficult to recognize novel images of that object later. Furthermore, for any novel image there should be one object model that unambiguously fits it best. When the generative process is a poor one, $P(I \mid M)$ will be small for all models. If even the best models have low probability, there may not be a reliable difference between the likelihood of the correct and incorrect models. As a result recognition performance will almost certainly suffer.

In the CFR framework, the likelihood of an image is dependent on the particular features that are available. If none of the features fit a particular training image well, there will be no object model that will make this image likely. A different, more appropriate, set of features must be used to model this image. Good features are those that can be used to form likely models for an entire set of training images. In order to insure that CFR will be able to model a wide variety of object types, an automatic technique for finding good features is a necessity. We will present a such a technique that is based on the principle of maximum likelihood.

We are given a sequence of images, $\{I(t)\}$ (though t is simply an index into the sequence, in the next section we will explicitly assume that t is in fact time). If the probabilities of the these images are independent, then the maximum likelihood estimate for f_i is found by maximizing the likelihood:

$$\ell = \prod_t P(I(t) \mid d_i(t), l_i(t), f_i) \ . \tag{19}$$

Since we do not know $d_i(t)$ and $l_i(t)$, we can either integrate them out or choose the best:

$$\begin{aligned} \ell &= \prod_t P(I(t) \mid f_i) = \prod_t \sum_{d_i(t), l_i(t)} P(I(t) \mid d_i(t), l_i(t), f_i) P(d_i(t), l_i(t)) & (20) \\ &\approx \prod_t \sum_{d_i(t)} \max_{l_i(t)} P(I(t) \mid d_i(t), l_i(t), f_i) P(d_i(t), l_i(t)) \ . & (21) \end{aligned}$$

In many cases it is most convenient to maximize $\log(L)$ (which has the same maximum as L):

$$\log(\ell) \approx \sum_t \sum_{d_i(t)} \max_{l_i(t)} \left[\log(P(I(t) \mid d_i(t), l_i(t), f_i)) + \log(P(d_i(t), l_i(t)))\right]. \tag{22}$$

Since computing the maximum of ℓ can be quite difficult, we will resort to gradient based maximization. Starting with an initial estimate for f_i we compute the gradient of ℓ with respect to f_i, $\nabla_{f_i}\ell$, and take a step in that direction. While this may seem like a complex calculation it has a simple implementation:

- For each $I(t)$ find the $l_i(t)$ that maximizes $P(I(t) \mid d_i(t), l_i(t), f_i)$. This is implemented much like a convolution where the point of largest response is chosen.
- Extract $S(I(t), l_i(t))$ for each time step.
- Compute the gradient of ℓ with respect to f_i. (For notational simplicity we have dropped the functional notation for time dependence, $l_i(t)$ and $d_i(t)$. These variables are still functions of t however.):

$$\nabla_{f_i}\ell = \sum_t \frac{\sum_{d_i(t)} \max_{l_i(t)} \frac{d}{df} P(I(t) \mid d_i(t), l_i(t), f_i)}{\sum_{d_i(t)} \max_{l_i(t)} P(I(t) \mid d_i(t), l_i(t), f_i) P(d_i(t), l_i(t))} \tag{23}$$

$$= \frac{N(S(I, l_i), f_i, \Sigma_i)}{N(S(I, l_i), f_i, \Sigma_i) + \frac{1}{R^{n_i}}}\ 2\ [S(I, l_i) - f_i]\ . \tag{24}$$

See Equations 4 and 5 for the definition of the probability of an image given a feature. The above equation can be written more simply as,

$$\nabla_{f_i}\ell = \sum_t \Gamma(t)\ [S(I(t), l_i(t)) - f_i]\ , \tag{25}$$

which is a weighted combination of differences.

- Take a small step in the direction of the gradient $f_i^{new} = f_i^{old} + \alpha \nabla_{f_i}\ell$.
- Repeat until f_i stabilizes.

4.1 Learning Useful Features

This algorithm can be used to "learn" a set of features that model a class of images well. There is nothing however that insures these feature will be well suited to the problem of visual object recognition. Nothing encourages the features to be *stable* or *distinct*. Two very similar views of an object may be exquisitely well modeled by two very different feature representations. In order to support general object recognition CFR must use features that where similar views of an object are represented with similar, if not identical, models.

Optimally, though perhaps unachievable, the CFR feature representation of an object should be constant across changes in pose – a constant representation is certainly stable. This condition can be encouraged with the following approach. Take a variety of different views of an object, $\{I(t)\}$, and attempt to maximize the likelihood, ℓ, where some set of $d_i(t)$'s are always 1. This is somewhat different from the previous approach where $d_i(t)$ was unknown. It has several disadvantages. It assumes that it is possible to build an object representation that is invariant to pose – a very difficult if not impossible task. Furthermore, it can be difficult to determine apriori which features should belong to which objects.

While attempting to learn a constant CFR feature representation may be impractical, learning a stable representation is not. One useful definition of "stable" is that as an object slowly changes pose, large changes in representation are rare while small changes in representation are more common. One can formulate this in a way that is very similar to a smoothness prior that is common in regularization theory (Poggio, Torre and Koch, 1985).

Assuming that $\{I(t)\}$ is a sequence of images of an object smoothly varying in pose, we can express our bias toward stability in the following way[4]:

$$P(D_i(t) = 1|D_i(t-1) = 1) = P(D_i(t) = 1|D_i(t-1) = 1) = p_c \tag{26}$$

$$P(D_i(t) = 0|D_i(t-1) = 1) = P(D_i(t) = 1|D_i(t-1) = 0) = p_t \tag{27}$$

$$P(L_i(t) = v|L_i(t-1) = w, D_i(t-1) = 1, D_i(t) = 1) = N(v, w, \Sigma) \tag{28}$$

$$P(L_i(t) = v|L_i(t-1) = w, D_i(t-1) = 0 \mathrm{OR} D_i(t) = 0) = 1/W \quad , \tag{29}$$

The first two equations determine the prior probability that $D_i(t)$ will remain constant through time (p_c is the probability that D_i will remain constant and p_t is the probability that D_i will transition). The third equation determines how probable changes in location are (changes in location are distributed as a gaussian around the previous location). Both (Földiák, 1991) and (Becker, 1993) have suggested that temporal continuity may serve as a mechanism for learning object identity.

These difference sources of information about the likelihood of a feature representation can then be combined. The new "stable" form of the likelihood of image formation is:

$$\sum_t \sum_{d_i(t), d_i(t-1)} \max_{l_i(t), l_i(t-1)} \left(\hat{P}\right) \tag{30}$$

where

$$\hat{P} = \left(\begin{array}{ll} & \log P(I(t) \mid d_i(t), l_i(t), f_i) \\ + & \log P(d_i(t), l_i(t)) \\ + & \log P(D_i(t) = d_i(t)|D_i(t-1) = d_i(t-1)) \\ + & \log P(L_i(t) = l_i(t)|L_i(t-1) = l_i(t-1), D_i(t-1) = d_i(t-1), D_i(t) = d_i(t)) \end{array} \right) \tag{31}$$

[4]These probabilities are implicitly conditioned on the fact that $I(t)$ and $I(t+1)$ contain images of the same object in a similar pose.

Following a very similar algorithm to the one detailed above, we can compute the derivative of the likelihood of an image sequence. The gradient

$$\nabla_{f_i} \log(\ell) = \sum_t \Gamma(t)\Gamma_d(t)\ \Gamma_l(t)\ [S(I(t), l_i(t)) - f_i] \tag{32}$$

is again a weighted sum of differences. We are currently exploring the possibility of optimizing the trajectory of D_i and L_i across longer periods of time. In that case the appropriate formulation is as a hidden markov model.

The priors we have added embody the assumption that the image sequence contains slowly varying images of one object followed by slowly varying images of some other object. The sequence cannot contain a hodge-podge of images collected from different objects.

5 Oriented Energy and Feature Matching

While CFR may be an effective technique for representing images and learning features, its generalization performance is very dependent on the pixelated input representations used. An effective representation should be insensitive to the foreseeable variations observed in images, while retaining all of the necessary information required for recognition. For example, the image pixels of a an object will vary rapidly as both the illumination and pose of the object changes. In order to insure good generalization the pixelated representations used should be insensitive to these changes.

Sensitivity to pose is directly related to the spatial smoothness of the pixelated representation. If the pixelated images are very smooth, pixel values will change slowly as pose is varied. A effective representation for recognition should enforce pixel smoothness without removing the information that is critical for discriminating features. This seems like a conflicted goal. On one hand we want to smooth, attenuating high-frequencies and reducing information. On the other we want to preserve information about higher frequencies to preserve selectivity. Oriented energy separates the smoothness of the representation from the frequency sensitivity of the representation. High frequency information can be preserved in a way that allows for positional flexibility.

The calculation of of oriented energy proceeds in two stages: linear and non-linear. First the input image is convolved with two Gabor functions that are orthogonal and oriented (this is the linear part). These filters share the same spatial window, orientation and frequency characteristics. They vary only in phase (see Figure 9). Second, the sum of the squares of the outputs of these filters are collected into an image (the non-linear part). Since the filters form a quadrature pair, the result can be viewed as an energy. Gabor filters are localized both in frequency and in space. The convolution output gives us information both about the frequencies in the input image and their locations. We could use the outputs of these filters directly as an alternative representation of the image. Since the filters are band-pass, correlation in the outputs

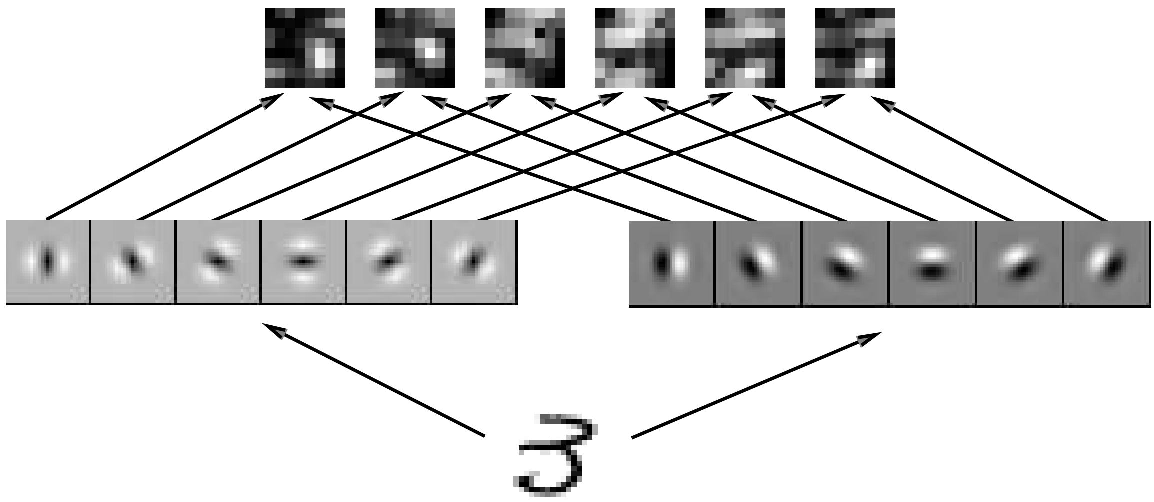

Figure 9: This figure contains a diagrammatic depiction of the computation of oriented energy. On the bottom is the input image, in this case a 3. Oriented energy is computed by two banks of filters, an odd bank shown on the right, and and on even bank shown on the left. The image is convolved separately with each of the 12 filters. The resulting 12 images are then squared. The 6 resulting maps are constructed by summing the squared outputs of the even and odd filters that have the same orientation.

may be larger than in the original image. But, by squaring and summing the outputs we get a measure of the energy in the band-pass frequencies that is invariant to phase. A maximum in this energy corresponds the the classical definition of an intensity edge[5]. The energy image has the frequency signature of the window function, a Gaussian. Since a Gaussian is essentially a low-pass filter, the resulting energy image has more spatial smoothness than the input image.

We can now return our attention to the features shown in Figures 6 and 7. Oriented energy allows for a selective description of the face, without being overly constraining about the location of important properties. Noses are strongly vertical pixels surrounded by the strongly horizontal pixels of the eyebrows. Figure 10 shows another feature which responds strongly to the right eye of a head. In this feature both the eye, the eyebrow and the right hairline seem to be represented. It is important to note that the physical structure to which this complex feature responds was not enforced by any teacher. CFR's feature learning procedure settled on the right eye because it is stable.

Another major aspect of image variation is illumination. The value of a pixel can change significantly with changes in lighting. We will assume that for the most part lighting varies slowly across a scene. As a result a large portion of the variation

[5]In fact Freeman and Adelson used oriented energy as an input to a Canny edge detector and found that performance was significantly improved (Freeman and Adelson, 1991) (Canny, 1986).

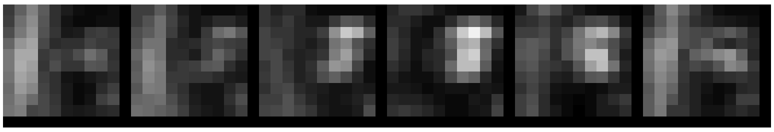

Figure 10: A feature which learned to respond to the right eye of a head. (Note: due to production difficulties the white boarder between the oriented energy maps is not printed.)

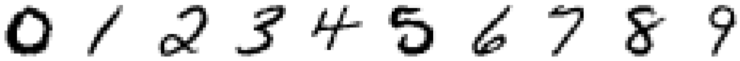

Figure 11: Ten example digits.

can be modeled locally, for the purposes of feature matching, as either an additive or multiplicative effect. Fortunately, oriented energy is already invariant to additive offset. Multiplicative effects can be eliminated by normalizing the length of both $S(I, l_i)$ and f_i before the comparison is made.

6 Experiments

In many ways this paper contains a collection of related insights about object recognition: i) oriented energy is an effective means of representing images, ii) features can be learned that are stable and iii) images are well represented with complex features. Let us address these issues in order.

For handwritten digits, oriented energy is a more effective representation than the pixels of an image (see Figure 11 for example digits). We constructed a nearest neighbor classifier, which is the simplest possible feature based recognizer. It works by classifying each novel digit to the class of the closest training digit. The training set had 75 examples of each digit as did the completely separate test set. Using the pixels of the images directly performance was 81%. Using an oriented energy representation, and no other changes, performance jumped to 94%.

Complex features can be learned from the data in an unsupervised fashion. The features shown in Figures 7 and 10 are examples of such features. The locations estimated in Figure 5 are typical of what can be expected for features learned from motion sequences.

We have tested CFR on a number of different recognition tasks. We obtained two databases of real objects: a set of images of five small objects taken under controlled conditions from Shree Nayar of Columbia (see Figure 12) and a database of ten people from David Beymer of the MIT AI Lab (see Figure 13). The object database contains 72 different views of each object, 9 of which we used for training. The face dataset contains 20 views of each face, 15 were used for training and 5 for testing.

We tested CFR-MEM and CFR-DISC on both these datasets. In all cases we used

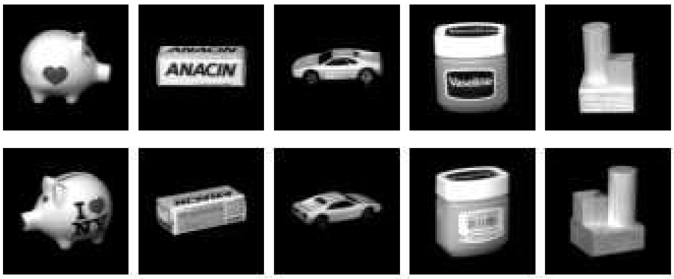

Figure 12: These are example images from the object dataset.

Figure 13: These are example faces from the face dataset.

20 features. The datasets contained many views that are fairly close together in pose space. This allowed us to treat them as if they were a motion sequence. The features were trained to maximize the likelihood of these sequences. The initial estimates for the features were snap-shots randomly chosen from the training set. For the face data we ran the classification experiments both with the initial random features and with the trained features.

	Chance	CFR-MEM Random Features	CFR-DISC Learned Features	CFR-MEM Random Features	CFR-DISC Learned Features
Objects	20		99		99
Faces	10	70	90	90	95

These results are about as good as the results that Nayar reports on his own data, but not as good as the results that Beymer reports on his data (Murase and Nayar, 1993) (Beymer, 1993). In general CFR is very easy to use. For the most part CFR runs without requiring any intervention. The features are learned, the models are created and images are recognized without supervision. The exact same code runs on both the objects and the faces. Once trained, CFR is quite efficient taking no more than a couple of seconds to recognize each image.

CFR has been tested on a few totally natural images. In these cases there has been no control of lighting, and little control of pose and camera parameters. In

Figure 14: This preliminary result demonstrates that CFR can be used both in complex cluttered scenes and for class recognition tasks. The system was trained on 15 views of 10 different people. It was then asked to identify which regions of the images were likely to have a face. These regions were labeled with a white square. Each square corresponds to a region that is about 2 times the size of the largest head in the image.

one experiment we tested class recognition. The goal of this experiment was to take a novel image and label those regions of the image that may contain a face. The training data included as positive examples the face data mentioned above, and as negative examples a variety of random backgrounds taken from real images. The test image was broken up into overlapping regions, each of which was labeled as either containing a face or not. Each region was about twice as large as the largest face in the training set. In the test image shown (see Figure 14) a white square is placed at the center of every region in which the CFR estimate for the probability of "face" was larger than the probability of "background". This test image was taken with a different camera and under different conditions from the training set. None of the people in the test image are in the training set.

7 Related Work

A complete review of related work in object recognition would be far beyond the scope of this paper. Certainly the concept of a feature representation for images is not new. The majority of the related work falls into two disjoint groups: techniques that use simple local features such as edges, and techniques that use complex global features. While edge based techniques have proven widely successful, we believe that they have drawbacks. We feel that CFR, though in its infancy, may open paths toward the recognition of more general classes of objects. In addition, with appropriate features, CFR should be more efficient than a brute force matching of simple features.

Techniques that use complex global features come in a wide variety of types. Recent examples include color histograms (Swain and Ballard, 1991), shape measures such as those proposed in (Sclaroff and Pentland, 1995) or monolithic neural networks such as those proposed by (Le Cun et al., 1989). These techniques are distinguished because they are capable of using many different types of information, like color and texture. They do, however, share a sensitivity to clutter and frequently assume that the object is segmented from the background. In fact the very concept of "global" pre-supposes that the extent of the object is known. We hope that CFR combines the best properties of both the global and local techniques.

Recently, (Rao and Ballard, 1995) have proposed that a type of pre-processing similar to oriented energy be used on images before they are matched to models. Though there are significant differences, Rao and Ballard's representation for an object model is much like CFR's representation of a single pixel of a single feature. Recognition then proceeds in a manner similar to the CFR-memorize algorithm. It is our hope that the formal model for CFR can be applicable to this work. In addition our insights on feature learning may prove useful in their construction of object models.

8 Conclusion

This paper has presented a formal framework within which two different object recognition algorithms have been derived. This framework is constructed on the insight that images are well represented as collections of complex local features. Since this framework is dependent on the quality of the features used, we have additionally derived an algorithm that is capable of automatically learning a set of features which are appropriate for object recognition. This novel algorithm for learning features requires no outside supervision. From random initial hypotheses ineffective features are discarded, and effective features are refined. Finally recognition performance is improved by pre-processing the input images so that intensity changes at different frequencies and orientation are enhanced.

References

Ballard, D. H. (1981). Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 3(2):836–840.

Becker, S. (1993). Learning to categorize objects using temporal coherence. In Hanson, S. J., Cowan, J. D., and Giles, C. L., editors, *Advances in Neural Information Processing*, volume 5, Denver 1992. Morgan Kaufmann, San Mateo.

Beymer, D. (1993). Face recognition under varying pose. AI Memo 1461, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Bolles, R. C. and Cain, R. (1982). Recognizing and locating partially visible objects: The local–feature–focus method. *International Journal of Robotics Research*, 1(3):57–82.

Brunelli, R. and Poggio, T. (1992). Face recognition: Feature versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052.

Canny, J. F. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698.

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley and Sons.

Földiák, P. (1991). Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200.

Freeman, W. T. and Adelson, E. H. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906.

Grimson, W. E. L. and Lozano-Perez, T. (1984). Model-based recognition and localization from sparse range or tactile data. *International Journal of Robotics Research*, 3(3):3–35.

Kandel, E. and Schwartz, J. (1985). *Principles of Neural Science*. Elsevier, New York, second edition.

Le Cun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., and Jackel, L. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551.

Marr, D. and Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London*, B(207):187–217.

Murase, H. and Nayar, S. K. (1993). Learning and recognition of 3-d objects from brightness images. In *AAAI Fall Symposium Series Working Notes*. AAAI.

Poggio, T., Torre, V., and Koch, C. (1985). Computational vision and regularization theory. *Nature*, 317:314–319.

Rao, R. P. N. and Ballard, D. H. (1995). Object indexing using an iconic sparse distributed memory. In *Proceedings of the International Conference on Computer Vision*, pages 24–31, Cambridge, MA. IEEE, Washington, DC.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E. and McLelland, J. L., editors, *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, chapter 8. MIT Press.

Sclaroff, S. and Pentland, A. P. (1995). Modal matching for correspondence and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6):545–561.

Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, 7:11–32.

Wells III, W. M. (1991). MAP Model Matching. In *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, pages 486–492, Lahaina, Maui, Hawaii. IEEE.