Pattern Rejection [*]

Simon Baker and Shree K. Nayar
Department of Computer Science, Columbia University, New York, USA

Abstract

The efficiency of pattern recognition is particularly crucial in two scenarios; whenever there are a large number of classes to discriminate, and, whenever recognition must be performed a large number of times. We propose a single technique, namely, pattern rejection, that greatly enhances efficiency in both cases. A rejector is a generalization of a classifier, that quickly eliminates a large fraction of the candidate classes or inputs. This allows a recognition algorithm to dedicate its efforts to a much smaller number of possibilities. Importantly, a collection of rejectors may be combined to form a composite rejector, which is shown to be far more effective than any of its individual components. A simple algorithm is proposed for the construction of each of the component rejectors. Its generality is established through close relationships with the Karhunen-Loéve expansion and Fisher's discriminant analysis. Composite rejectors were constructed for two representative applications, namely, appearance matching based object recognition and local feature detection. The results demonstrate substantial efficiency improvements over existing approaches, most notably Fisher's discriminant analysis.

1 Introduction

The efficiency of a pattern recognition algorithm becomes ever more vital as the number of pattern classes increases. Of particular importance is the growth rate of the recognition time as a function of the number of classes. Appearance matching based object recognition [Murase and Nayar 95] is one example application where the computational dependence upon the number of classes (objects) is the key to a real time solution. Another such application in computational vision is face recognition [Pentland et al. 94] [Sirovich and Kirby 87] [Turk and Pentland 91]. High efficiency also proves critical whenever the recognition algorithm must be applied a large number of times. This is the case in local feature detection [Nayar et al. 96], where the feature detector must be applied at every pixel in an image.

We propose a general theory that results in substantial efficiency improvements in both of the above scenarios. The central concept is that of *rejector.* A rejector is an algorithm that quickly eliminates a large percentage of the candidate classes (e.g. objects in recognition) or inputs (e.g. local image brightness values in feature detection). The notion of a rejector is fundamental since it leads to the following important observations:

1. The definition of correctness for a rejector is less constraining than that for a classifier (recognizer). As a result, rejectors can be constructed that are far more efficient than corresponding classifiers.
2. Although, in general, a single rejector does not provide a final solution to the pattern recognition problem, it significantly reduces the number of possible classes or inputs. Consequently, the recognizer can dedicate its computational resources to a much smaller number of candidates.
3. A collection of rejectors can be combined in a tree-like structure to form a much more effective rejector, which we term a *composite rejector.* At each node of the composite rejector is a simple rejector that is specifically tuned to the reduced subset of classes not eliminated by the previous rejector.
4. It is possible to analyze the performance of composite rejectors. For instance, we derive conditions that guarantee logarithmic time complexity in terms of the total number of classes.
5. We develop a very general algorithm for rejector construction based on a single assumption, namely, the *class assumption.* The generality of the class assumption is established through a close connection with the Karhunen-Loéve expansion [Fukunaga 90]. A relationship between the rejector construction technique itself and Fisher's discriminant analysis [Duda and Hart 73] is also shown.

We demonstrate the utility of pattern rejection via experiments on appearance matching based object recognition [Murase and Nayar 95] and feature detection [Nayar et al. 96]. First, we constructed a composite rejector for a widely used image database of 20 objects. Each object appears in a large number of poses and constitutes a single pattern class. The composite rejector is able to completely (and without error) discriminate between all 20 objects with an efficiency that is a significant improvement over currently used techniques. We empirically illustrate logarithmic growth in the time complexity with the number of objects. Further, we compare the composite rejector with Fisher's discriminant analysis and show rejection to be both substantially more efficient as well as more accurate. Finally, we constructed a composite rejector for the task of feature detection. The result is a very efficient method of preprocessing an image to identify pixels that truly deserve the application of a full-fledged feature detector, such as the one proposed in [Nayar et al. 96].

[*]This research was supported in parts by ARPA Contract DACA-76-92-C-007, by DOD/ONR MURI Grant N00014-95-1-0601, and by an NSF National Young Investigator Award.

1.1 Relationship with Previous Work

The recursive structure of the composite rejector constitutes a decision tree, (or more generally a directed acyclic graph). A survey of the pattern recognition literature that uses such a structure is beyond the scope of this paper, but a small selection is [Henrichon and Fu 69] [Payne and Meisel 77] [Weng 94]. For a brief discussion of decision trees see [Duda and Hart 73]. Connections can also be made between our results and the large body of work on computationally motivated nearest neighbor classifiers [Friedman et al. 77] [Bentley 80] [Yianilos 93]. Though the problem we address is similar, namely, efficient classification, our setting is more general.

A relationship can be established between our algorithm for rejector construction and Fisher's discriminant analysis [Fisher 36] [Duda and Hart 73]. In particular, our choice of rejection vectors in Section 3.3 will be seen to tend to maximize between-class scatter, while keeping within-class scatter fixed at a low level. The major difference between rejection theory and discriminant analysis is that discriminant analysis is presented as a single level of processing. On the other hand, a composite rejector has a hierarchical structure, which leads to superior performance. In particular, the relative performance is accounted for by the fact that each rejector in the composite rejector is individually constructed for a subset of classes which is as small as possible. Hence, the second and subsequent Fisher vectors can be regarded as suboptimal when compared to the rejection vectors used in the composite rejector. Weng [Weng 94] uses a similar hierarchical structure that takes advantage of this property.

Another important characteristic of our approach is the central role played by the pattern classes. Existing work which is concerned with efficiency either models the classes as collections of points, or studies partitions of a feature space. Hence, our complexity results are in terms of the number of classes, rather than the number of sample points. Importantly, this class-centered approach focuses attention on what we believe to be the key question: What properties must the pattern classes possess for recognition to be performed efficiently? The introduction of the class assumption is an attempt to answer this question.

2 Theory

2.1 Assumptions and Definitions

A pattern recognition problem is based on a finite set of measurements of an underlying physical process. Hence, we assume the existence of a *classification space*, $S = \Re^d$, where d is the number of measurements. Elements, $x \in S$, will be referred to as *measurement vectors*, or for convenience, *vectors*. Next, we assume the existence of a finite collection, $W_1, W_2, \ldots, W_n \subseteq S$ of *(pattern) classes*. The classes themselves are defined by the application in question. We will therefore assume that the classes are given to us *a priori*. We are now ready to define a classifier:

Definition 1 *A* classifier (or recognizer) *is an algorithm, ϕ, that given an input, $x \in S$, returns the class label, i, for which $x \in W_i$.*

A *rejector* is a generalization of a classifier in the sense that it returns a set of class labels. This set must contain the correct class, but may also contain others:

Definition 2 *A* rejector *is an algorithm, ψ, that given an input, $x \in S$, returns a set of class labels, $\psi(x)$, such that $x \in W_i \Rightarrow i \in \psi(x)$ (or equivalently $i \notin \psi(x) \Rightarrow x \notin W_i$).*

The name rejector comes from the equivalent definition: $i \notin \psi(x) \Rightarrow x \notin W_i$. That is, if i is not in the output of the rejector, we can safely reject the possibility that $x \in W_i$. We then define the *rejection domain* for W_i to be the set of all $x \in S$ for which i does not appear in the rejector output:

Definition 3 *If ψ is a rejector and W_i is a class, then the* rejection domain, *R_i^ψ, of rejector, ψ, for class W_i is the set of $x \in S$ for which $i \notin \psi(x)$.*

Then, the following important properties hold:

1. Subject to $R_i^\psi \cap W_i = \emptyset$, we are free to choose the rejection domains and still conform with the correct definition of a rejector. This freedom to choose rejection domains with "simple" decision boundaries is what allows rejectors to be efficient.
2. The rejector, ψ, may be used to reject $x \in W_i$ iff $x \in R_i^\psi$. Hence, we should aim to choose the rejection domains to be as large as possible. However, there is a trade-off between maximizing the size of the rejection domains, ensuring $R_i^\psi \cap W_i = \emptyset$, and using simple decision boundaries for efficiency.

2.2 Rejection Based Classifiers

Applying a rejector does not guarantee that we will be able to uniquely solve the pattern recognition problem. There may be more than one class in the output of the rejector. We deal with the potential ambiguity by adding a verification stage:

Definition 4 *A* verifier *for a class, W_i, is a boolean algorithm that, given an input, $x \in S$, returns the result $True$ if x is a member of W_i and $False$ otherwise.*

We form a *rejection-based classifier* by first applying a rejector, ψ, and then applying a verifier for each class label in the output of the rejector. Combining the results, we can easily classify the input. The efficiency of a rejection-based classifier can immediately be seen [Baker and Nayar 95] to be given by:

$$T_{av}(\phi^{rb}) = T_{av}(\psi) + E_{x \in S}(|\psi(x)|) \cdot T_{ver} \qquad (1)$$

where, $T_{av}(\phi^{rb})$ is the average run time of the rejection-based classifier, $T_{av}(\psi)$, is the average run time of the rejector, $E_{x \in S}(|\psi(x)|)$ is the expected cardinality of the rejector output, and T_{ver} is the run time of each of the verifiers (assumed to be the same for all verifiers). We now introduce a further definition:

Definition 5 *If ψ is a rejector, the* effectiveness *of ψ is* $\mathrm{Eff}(\psi) = \frac{E_{x \in S}(|\psi(x)|)}{n}$

Note that a small numeric value of $\mathrm{Eff}(\psi)$ corresponds to an "effective" rejector. Then, equation (1) shows that a rejection-based classifier will be efficient when, (a) rejection is *efficient*, and (b) rejection is *effective*.

2.3 Composite Rejectors

As we will see, constructing very efficient rejectors is relatively straightforward, but in some applications, the rejectors tend to be less effective than might be hoped for. However, applying a rejector results in a subset of classes, and so a smaller instance of the original classification problem. Recursively applying another rejector, tuned to the smaller subset of classes, should enable us to further narrow down the set of classes under consideration. Overall, we can expect a significant improvement in the combined effectiveness. This is the notion of a *composite rejector:*

Definition 6 *A* composite rejector, Ψ, *is a collection of rejectors, $\Psi = \{\psi_\imath : \imath \in \Im\}$, where $\Im$ is an index set for Ψ, and such that, (a) there is a rejector in Ψ designed for the complete set of classes, and (b) for any rejector, $\psi_\imath \in \Psi$, and any $x \in S$, either $\psi_\imath(x) = 1$ or there is a rejector in Ψ designed for $\psi_\imath(x)$.*

The composite rejector is laid out in the form of a directed acyclic graph. Each rejector, $\psi_\imath \in \Psi$, and the subset of classes for which it was designed, corresponds to a node in the graph. There is a directed edge from the node corresponding to $\psi_\imath$ to that corresponding to $\psi_\jmath$, if and only if there is a vector, $x \in S$, such that $\psi_\jmath$ was designed for the subset of classes, $\{W_i : i \in \psi_\imath(x)\}$. Then, the application of the composite rejector to a measurement vector corresponds to a path through the directed acyclic graph. At each node in the path, the associated rejector is applied and its output determines the edge that should be taken to leave the node.

2.4 Time Complexity

Intuitively, the recursive structure of the composite rejector leads us to expect logarithmic time complexity for the resulting rejection-based classifier. Sufficient conditions to prove such a result are as follows:

1. For all $\psi_\imath \in \Psi$, and for all $x \in S$, either $|\psi_\imath(x)| = 1$, or at least one class is eliminated by $\psi_\imath$.
2. With respect to the underlying *a priori* probability density function from which the vectors are drawn, the rejection domains are mutually independent.
3. The effectiveness of each of the rejectors is the same: $\forall \imath \in \Im$, $\mathrm{Eff}(\psi_\imath) = \mathrm{E}$, say.

Then, we can show (see [Baker and Nayar 95]) that a rejection-based classifier using such a composite rejector runs on average in time:

$$T_{av}(\phi^{rb}) \leq \lceil \log_{\mathrm{E}^{-1}} n \rceil \cdot T_{rej} + 2 \cdot T_{ver} \qquad (2)$$

where, T_{rej} is the run time of each of the rejectors (assumed constant), and n is the number of classes.

2.5 Space Complexity

A potential problem with the composite rejector is that the number of component rejectors may be as large as 2^n. To limit the growth in the size of Ψ, we must impose constraints on the design of each $\psi_\imath$. We require that: (a) for each $\psi_\imath \in \Psi$, the number of different possible outputs is two, (b) the two possible outputs are of equal cardinality, and (c) the intersection between the outputs consists of at most a fraction, $\epsilon \in [0, 1)$, of the classes for which that rejector was constructed. If we denote by $M(n)$, the maximum number of rejectors in Ψ that may be reached after, and including, the rejector constructed for a collection of n classes, it can be shown (see [Baker and Nayar 95]) that $M(n)$ is polynomial:

$$M(n) \leq n^{1/(1-\log_2(1+\epsilon))} - 1 \qquad (3)$$

It is not always possible to completely satisfy the three requirements stated above. However, the following design criteria may be used as guidelines while implementing each component rejector in Ψ: (a) avoid designs that produce a large number of outputs, (b) attempt to balance the output cardinalities, and (c) minimize the overlap between outputs.

3 A Rejector Construction Algorithm

We now assume that the norm of a measurement vector is unimportant for classification and restrict attention to the unit ball, $B = \{x \in S : \|x\|_2 = 1\}$.

3.1 The Class Assumption

Designing a rejector is equivalent to deciding on the rejection domains. Since for correctness we require $R_i^\psi \cap W_i = \emptyset$, the choice of the rejection domains depends heavily on the nature of the underlying classes. Hence, we make the following assumption (see Figure 1):

The Class Assumption *For each W_i, there exists a vector, $c_i \in S$, a linear subspace, $L_i \subseteq S$, and a threshold, $\delta_i \geq 0$, such that $\forall x \in W_i$, $\mathrm{dist}(x, c_i + L_i) \leq \delta_i$. Further we assume: (a) $\dim(L_i) \ll d$, and (b) $\delta_i \ll 1$.*

The class assumption is very general and is approximately equivalent to assuming that the K-L expansion results in a compact and accurate representation of the class. Suppose that M_i^k is the subspace spanned by the k most important K-L eigenvectors, and λ_i are the decaying K-L eigenvalues, then we have:

$$E_{x \in W_i}(\mathrm{dist}(x, E_{y \in W_i}(y) + M_i^k)^2) = \sum_{s=k+1}^{d} \lambda_s \approx 0. \quad (4)$$

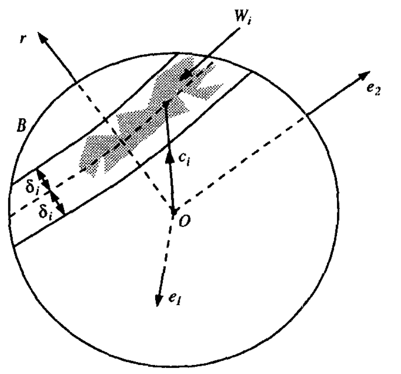

Figure 1: An illustration of the class assumption for a low dimensional example, $S = \Re^3$. The subspace, L_i, is the 2 dimensional subspace spanned by the vectors, $\{e_1, e_2\}$. Every vector in W_i can be approximated to within error, δ_i, by the linear combination of c_i and a vector in L_i.

Setting $c_i = E_{x \in W_i}(x)$, and $L_i = M_i^k$, we see that the difference between the class assumption and the K-L expansion is one of expected versus maximum value.

3.2 A General Purpose Rejector

Starting from the class assumption, we now derive a rejector and show that it may be computed efficiently. We begin by defining a *rejection vector*:

Definition 7 *Suppose the class assumption holds for the classes, $W_1, W_2, \ldots, W_n$. Then a* rejection vector *is a unit vector, $r \in B$, for which $r \perp \bigoplus_{i=1}^n L_i$.*

If r is a rejection vector, it follows immediately from orthogonality and the Cauchy-Schwarz inequality, that:

$$x \in W_i \;\Rightarrow\; |\langle r, x\rangle - \langle r, c_i\rangle| \le \delta_i \qquad (5)$$

Equation (5) means that the rejection vector projects each class, W_i, onto approximately a point, $\langle r, c_i\rangle$, which is then characteristic of the class. So long as the points, $\langle r, c_i\rangle$, are well separated, the intervals $[\langle r, c_i\rangle - \delta_i, \langle r, c_i\rangle + \delta_i]$ will not intersect much. Then, we can use equation (5) to discriminate[1] between the classes. Hence, we define a *derived rejector*:

Definition 8 *Given that the class assumption holds for the classes, $W_1, W_2, \ldots, W_n$, and that $r \in B$ is a rejection vector, then, we define the* derived rejector, *ψ_r by: $i \in \psi_r(x) \Leftrightarrow |\langle r, x\rangle - \langle r, c_i\rangle| \le \delta_i$*

The derived rejector may be implemented very efficiently as follows. After normalizing the input vector, x, we compute the projection with the rejection vector to give $\langle r, x\rangle$. Then, the set of class labels, i, for which $\langle r, x\rangle$ lies in the interval, $[\langle r, c_i\rangle - \delta_i, \langle r, c_i\rangle + \delta_i]$, can be computed with $\lceil \log_2(2n+1)\rceil$ comparisons and a lookup table. This is because the derived rejector is a precomputable piecewise constant function and finding the constant segment in which $\langle r, x\rangle$ lies takes $\lceil \log_2(2n+1)\rceil$ comparisons using binary search.

[1]There is no guarantee that we will be able to find a rejection vector, r, that completely separates a given pair of classes. This does not effect the usefulness of a derived rejector, since a rejector is intended to eliminate a large fraction of the classes, not necessarily every last one.

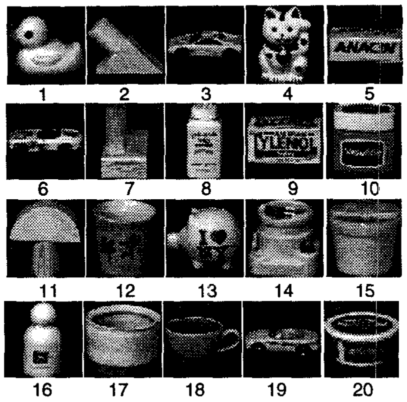

Figure 2: The 20 objects used in the recognition experiment, each of which constitutes a single pattern class. There are 72 images of each object, with each pair of images separated by 5° of pose. The data set is that used in [Murase and Nayar 95].

3.3 Choice of the Rejection Vector

We have seen that the derived rejector can be applied efficiently. The reason we can expect it to be effective is because we have quite some freedom in choosing the direction of the rejection vector, r. Thus far, r has only been constrained to lie orthogonally to $\bigoplus_{i=1}^n L_i$. Clearly, we should choose the rejection vector to be the one that spreads out the centers of the intervals, $[\langle r, c_i\rangle - \delta_i, \langle r, c_i\rangle + \delta_i]$, as much as possible. This will reduce the size of $|\psi_r(x)|$, and so tend to optimize the effectiveness of the derived rejector. If variance is used to measure the spread of the points $\langle r, c_i\rangle$, the best rejection vector to choose is the first Karhunen-Loéve eigenvector[2]

4 Implementation

While implementing the composite rejector, several practical issues must be addressed, including verifying the class assumption, estimating the thresholds, and avoiding the exponential growth in the space requirements. See [Baker and Nayar 95] for more details.

5 Example Applications

5.1 3D Object Recognition

We follow the appearance matching approach, first described in [Murase and Nayar 95]. Object recogni-

[2]This choice of rejection vector is closely related to Fisher's discriminant analysis [Duda and Hart 73]. By working orthogonally to $\bigoplus_{i=1}^n L_i$, we are limiting the within-class scatter of each class. Spreading out the points $\langle r, c_i\rangle$, maximizes the between-class scatter. The important difference, however, is that the derived rejector defers difficult decisions to subsequent rejectors which are more tuned to the reduced subset of classes.

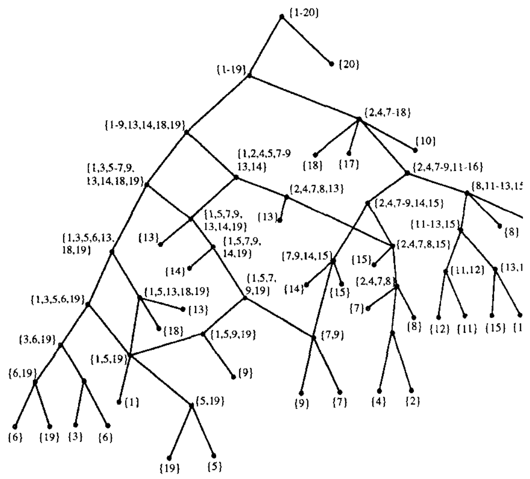

Figure 3: A representation of the composite rejector. Each interior node denotes a single rejector, and is labeled with the set of objects that it is designed to act on. At each node, only one dot product and a couple of comparisons need to be performed. Each leaf denotes a possible output of the composite rejector.

tion is reduced to pattern recognition by first segmenting the object and then scale normalizing it to an image of size 128×128 pixels. The image is then treated as a 16,384 dimensional measurement vector in the classification space, S, by reading the pixels in a raster scan fashion. Finally, the vector is intensity normalized to lie on the unit sphere, B.

The data set that we used (see Figure 2) consists of 20 objects (classes). It contains 72 images of each object separated by 5^o intervals of pose. The images were divided into a training and a test set each comprising 36 images of every object. The training set is then treated as samples of the classes and used to implement the composite rejector, a representation of which is presented in Figure 3. As it happens, every leaf of the composite rejector contains a single class, hence the composite rejector can fully discriminate between the 20 objects. (We would have regarded the rejector as successful even if each leaf had contained 2-3 objects.)

We found that the composite rejector responded 100% correctly for both the training and test sets. We calculated the average number of rejectors which the composite rejector applies to be just 6.43, based on the assumption that each image in the data set is equally likely to appear. Since the time taken at each node is essentially the cost of one inner product (convolution), the efficiency compares very favorably with the results obtained by Murase and Nayar [Murase and Nayar 95]. Their implementation required 20 inner products, followed by a sophisticated search procedure.

Next, we investigated the growth rate of the number of rejectors required as a function of the number of classes. The results, in Figure 4, validate the hypothesized logarithmic growth in time complexity.

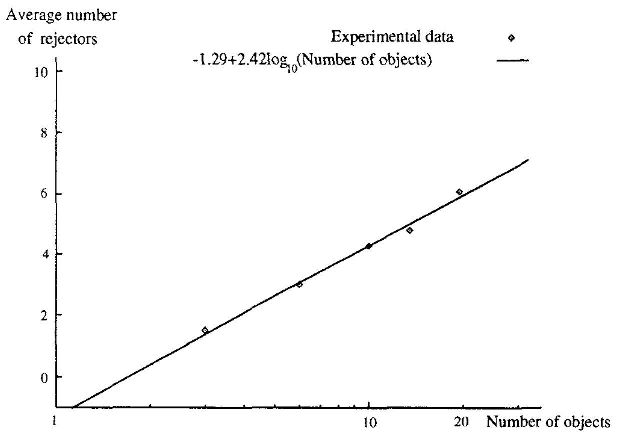

Figure 4: A graph of the number of objects against the average number of simple rejectors required to completely discriminate between those objects. The graph is plotted using a log scale on the abscissa, implying a logarithmic growth rate in the time complexity.

Using the same image database, we compared the performance of the composite rejector against that of Fisher's discriminant analysis [Fisher 36]. Again, we followed the same test procedure, namely, setting aside half of the data, and using the other half to construct the classifier. Then, we constructed Fisher spaces [Duda and Hart 73] of different dimensionality. In Fisher space the classes consist of tight clusters, which we model as multivariate normal distributions. We computed the mean and covariance matrix of each of these distributions. Then, each measurement vector was classified by finding its closest cluster, i.e. the cluster whose mean is closest to the vector. We used both the Mahalanobis and Euclidean distances.

Figure 5 shows the results for the combined performance on the training and test sets. Even for the Mahalanobis measure, the classification results are not perfect. In fact, the highest correct classification rate of 96.6% was attained for dimension 19. This compares poorly with the 100% classification obtained by the composite rejector, that required an average of just 6.43 rejection vectors. The main reason for the rejector's superior performance is its hierarchical structure. The rejector used at each step is individually constructed for the classes it seeks to distinguish between.

5.2 Local Feature Detection

We constructed a composite rejector for a feature detector of the type proposed in [Nayar et al. 96]. (The details of the implementation may be found in [Baker and Nayar 95].) The output of the composite rejector is used as input to the feature detector, and consists of pixels at which further consideration is worthwhile. Although the technique is applicable to general parametric features, we only have space to display our results (see Figure 6) for edge detection.

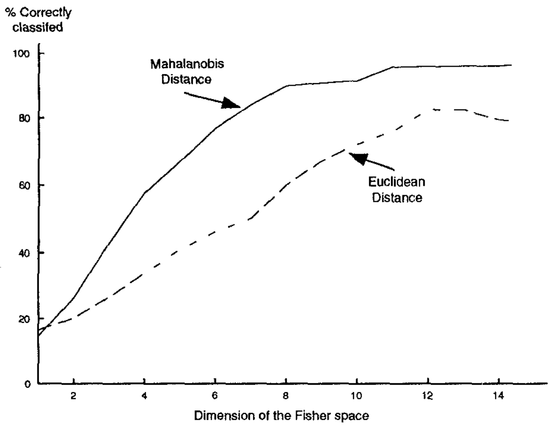

Figure 5: Results of applying Fisher's discriminant analysis to the data set in Figure 2. On the abscissa we plot the dimension of the Fisher space used, and on the ordinate the percentage of test images correctly classified. The peak performance is 96.6%, and to reach this 19 discriminant vectors are needed. In contrast, the composite rejector gives perfect (100%) classification with just 6.43 rejection vectors. Hence, by both measures, efficiency and robustness, the composite rejector outperforms discriminant analysis.

6 Discussion

Our goal has been to introduce a computational theory of pattern rejection. In this respect we have made considerable progress. We have introduced the notion of a composite rejector, developed algorithms to construct one, and demonstrated superior accuracy and efficiency over commonly used pattern recognition algorithms.

Pattern rejection works because, on average, given a large number of classes, discriminating between a randomly chosen pair of them is easy. Similarly, when applying a recognition algorithm a large number of times, very often the problem instance does not lie close to the decision boundary and so classification can be performed with almost no computational effort.

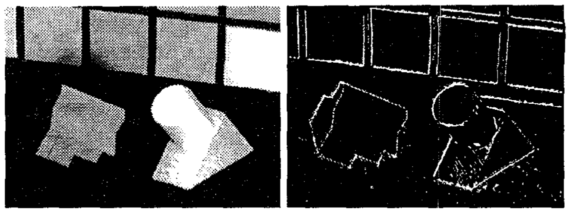

Figure 6: The edge rejector applied to a real image. The output on the right consists of those pixels which our rejection algorithm has quickly decided are worthy of further consideration. On average 1.81 rejectors (each corresponding to a convolution) were applied at each pixel.

References

[Baker and Nayar 95] S. Baker and S.K. Nayar, "A Theory of Pattern Rejection," *Columbia University Technical Report,* CUCS-013-95, 1995.

[Bentley 80] J.L. Bentley, "Multidimensional divide-and-conquer," *Communications of the ACM,* 23:214–229, 1980.

[Duda and Hart 73] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis,* John Wiley & Sons, 1973.

[Fisher 36] R.A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics,* 7:179–188, 1939.

[Friedman et al. 77] J.H. Friedman, J.L. Bentley, and R.A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Transactions on Mathematical Software,* 3:209-226, 1977.

[Fukunaga 90] K. Fukunaga, *Statistical Pattern Recognition,* Academic Press, 1990.

[Henrichon and Fu 69] E.G. Henrichon and K-S. Fu, "A Nonparametric Partitioning Procedure for Pattern Classification," *IEEE Transactions on Computers,* 18:614–624, 1969.

[Murase and Nayar 95] H. Murase and S.K. Nayar, "Visual Learning and Recognition of 3D Objects from Appearance," *International Journal of Computer Vision,* 14:5–24, 1995.

[Nayar et al. 96] S.K. Nayar, S. Baker, and H. Murase, "Parametric Feature Detection," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* San Francisco, 1996.

[Payne and Meisel 77] H.J. Payne and W.S. Meisel, "An Algorithm for Constructing Optimal Binary Decision Trees," *IEEE Transactions on Computers,* 26:905–916, 1977.

[Pentland et al. 94] A.P. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," In *Procedings of IEEE Conference on Computer Vision and Pattern Recognition,* pages 84–91, 1994.

[Sirovich and Kirby 87] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America,* 4:519–524, 1987.

[Turk and Pentland 91] M.A. Turk and A.P. Pentland, "Face recognition using eigenfaces," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* pages 586–591, 1991.

[Weng 94] J. Weng, "SHOSLIF: The Hierarchical Optimal Subspace Learning and Inference Framework," *Michigan State University Technical Report,* CPS 94-15, 1994.

[Yianilos 93] P.N. Yianilos, "Data Structures and Algorithms for Nearest Neighbor Search in General Metric Spaces," In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms,* pages 311–321, 1993.