HyperBF Networks for real object recognition

R. Brunelli[1], **T. Poggio**[2,1]
[1] Istituto per la Ricerca Scientifica e Tecnologica
I38050 Povo, Trento, ITALY
[2]Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, USA

Abstract— **Even if represented in a way which is invariant to illumination conditions, a 3D object gives rise to an infinite number of 2D views, depending on its pose. It has been recently shown ([6]) that it is possible to synthesize a module that can recognize a specific 3D object from any viewpoint, by using a new technique of learning from examples, which are, in this case, a small set of 2D views of the object. In this paper we extend the technique, a) to deal with real objects (isolated paper clips) that suffer from noise and occlusions and b) to exploit negative examples during the learning phase. We also compare different versions of the multilayer networks corresponding to our technique among themselves and with a standard Nearest Neighbor classifier. The simplest version, which is a Radial Basis Functions network, performs less well than a Nearest Neighbor classifier. The more powerful versions, trained with positive and negative examples, perform significantly better. Our results, which may have interesting implications for computer vision despite the relative simplicity of the task studied, are especially interesting for understanding the process of object recognition in biological vision.**

1. Introduction

Shape-based visual recognition of 3D objects may be solved by first hypothesizing the viewpoint (e.g., using information on feature correspondences between the image and a 3D model), then computing the appearance of the model of the object to be recognized from that viewpoint and comparing it with the actual image ([1], [2], [16], [3], [4]). Most recognition schemes developed in computer vision over the last few years employ 3D models of objects. Automatic learning of 3D models, however, is in itself a difficult problem that has not been much addressed in the past and which presents difficulties, especially for any theory that wants to account for human ability in visual recognition.

Recently, recognition schemes have been suggested that, relying on a set of 2D views of the object instead of a 3D model ([5], [21], [6]), offer a natural solution to the problem of model acquisition. In particular, Poggio and Edelman ([6]) have argued that for each object there exists a smooth function mapping any perspective view into a "standard" view of the object and that this multivariate function may be approximatevely synthesized from a small number of views of the object. Such a function would be object specific, with different functions corresponding to different 3D objects. Since synthesizing an approximation to a function from a small number of sparse data – the views – can be considered as learning an input-output mapping from a set of examples ([8], [15]), Poggio and Edelman used a scheme for the approximation of smooth functions which is equivalent to a class of multilayer networks called Regularization Networks and, in their more general form, HyperBasis functions. For each 3D object there exist a small network, which is "learned" directly from a small set of perspective views of the object. They demonstrated the successful performance of such a scheme using computer simulated 3D wireframe objects similar to paperclips. Their experiments assumed that the object had been isolated from the background and that features (such as the specific corners or angles between the segments) had been extracted and matched to the corresponding features of the model views. Furthermore, their data were noise-free and without any occlusions.

In this paper we extend their technique and experiments to more realistic situations. Our ultimate goal is to implement a system for the recognition of human faces by applying the HyperBF technique to view vectors computed from the image by extracting features such as the position of the eyes and mouth and the color of the hairs.

Real 3D objects introduces several difficulties, namely the presence of noise in the feature data, the ignorance of the correspondence between the features of different views of the same object and finally the necessity to use incomplete feature vectors (due to the presence of occlusions and/or to the inability to recover correctly some of the objects features). It seems reasonable to limit, at least in a first step, all of these difficulties to the recognition phase: the learning phase is supervised and uses "good" example views, where the problem of correspondence has been removed and the noise reduced.

The main result of the paper is that the Hyperbf technique, suitably modified, can deal successfully with the problems of noise, occlusions and missing correspondences, at least for the simple 3D objects we consider here. One of the most useful and interesting of our extensions of the techique is the use of negative examples in the training, that is in the model acquisition, phase.

The plan of the paper is as follows. The first section gives a brief review of the simple RBF technique. We then describe the experiments and compare the performance of different version of the algorithm (including performance of a standard Nearest Neighbor classifier). The more general HyperBF network is then introduced and characterized in terms of experimental performance.

2. Radial Basis Functions

Radial Basis Functions can be regarded as a special case of Regularization Networks introduced in [8] as a general approximation technique that can be used in problems of learning from examples.

A scalar function can be approximated, given its value on a sparse set of points $\{\bar{x}_i\}$, by an expansion in radial functions:

$$F(\bar{x}) = \sum_{i=1}^{N} c_i h(\| \bar{x} - \bar{x}_i \|) \qquad (1)$$

where $\| \cdot \|$ represents the usual Euclidean norm. The computation of the coefficients c_i rests on the invertibility of matrix $\mathbf{H}_{ij} = h(\| \bar{x}_i - \bar{x}_j \|)$ which has been proved (see Micchelli [17]) for functions such as:

$$h(r) = e^{-(\frac{r}{c})^2} \qquad (2)$$
$$h(r) = (c^2 + r^2)^{\alpha}, \alpha < 1 \qquad (3)$$

It is possible to use fewer radial functions than examples, i.e. data points. The resulting overconstrained system can be solved in a least square way under the conditions of Michelli's theorem and proves to be useful when many examples are available.

Poggio and Girosi ([8], [15]) have shown that the RBF technique is a special case of the regularization approach to the approximation of multivariate functions. In the regularization approach one seeks the approximating function which is closest to the data and smoothest, according to an appropriate criterion. The RBF technique described above, which is the simplest version of the HyperBF scheme described later, was used in the experiments described in the next section.

3. Experimental setup

The objects used in the experiments were five paper clips, randomly generated of the same length, rendered through ray-tracing techniques (see Fig. 1). The small number of objects used must be taken into account when the experimental results are considered. The small difference in performance among the different techniques is expected to increase if the number of objects to be classified increases. The clip was first isolated from the background ([18]) and the resulting binary image was skeletonized ([19]), giving essentially a line drawing. A polygonal approximation routine identified line segments which were then used to reconstruct the clip using an A* search ([18]). The same algorithms were applied to a real clip (images were taken with a CCD camera) and proven to perform equally well. The reconstructed clips were used to simulate different degrees of occlusions by assigning to each clip a finite radius (the bigger the radius the higher the occlusion percentage).

For each clip a set of 80 views were available among which the learning and testing sets could be chosen. The attitude was restricted to one octant of the viewing sphere. Each clip, having six vertices, was described as a vector of twelve coordinates, representing the position on the image plane of the vertices (expressed in the barycentric coordinate system identified by the vertices to remove translational dependency). The use of the (x, y) coordinates of the vertices on the image plane is one of the many possible choices of features. For instance, the angles between the clip segments could have been used ([6]). Broadly speaking, every feature that can be cast into numerical form and that maps smoothly into the output of the network, may be used in the recognition process.

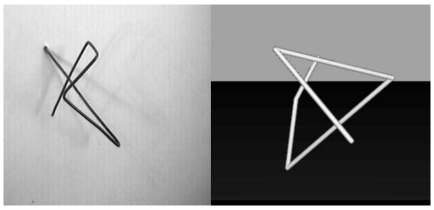

Fig. 1. A real clip (left) and a ray traced clip (right)

4. Experiments

The purpose of the experiments was to test the recognition performance of several strategies. For each rendered clip, from a set of 80 views, a learning set of given cardinality and a testing set of fixed cardinality (10 views) were extracted. The problem of feature correspondence was constrained by the nature of the objects. The clips could be present in the learning set in one of the two natural orientation (the reconstruction algorithm did not fix this ambiguity). In the recognition phase each clip was used in the two ways attempting recognition and choosing the best result. The order of the vertices was assumed to be correct but for this ordering ambiguity. The output representation used for recognition was the characteristic function of the clips:

$$F_i(\bar{x}) = \begin{cases} 1 & \text{if } \bar{x} \in P(C_i) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $P(C_i)$ represents the space spanned by the perspective projections of the i-th clip. The synthesis of the characteristic function from the learning set is realized by what will be referred to as *RBF module*. The response of a module to an input vector (a clip to be classified) is the value of the RBF expansion at the given point, clipped to the interval $[0, 1]$. The clip is then assigned to the module with the highest response.

Several groups of experiments have been performed:

- using only positive examples and complete vertex information;
- using only positive examples and occluded clips;
- using positive and negative examples with complete vertex information.

Positive examples are views of the correct clip and negative examples are views of other clips (therefore incorrect clips for the module under training).

In the first group of experiments the number of centers in the RBF expansions matched that of the available positive examples for the given clip. The performance reflects how well the RBF scheme works with noise in the feature vector and in the specific task. As we mentioned before, real 3D objects exhibit occlusions (topological absence of features), unrecovered features (defective feature extraction) and correspondence problems. The first two characteristics require the use of incomplete feature vectors. In the "vanilla" version of RBF described in section 2, the radial basis functions must be completely specified with their parameters before training. For Gaussian radial basis functions, the value for σ must be chosen. Let $\{\bar{x}_i\}_I$ be the learning set. To each example $\bar{x}_i$ a nearest neighbor can be associated, i.e. an example $\bar{x}_j \neq \bar{x}_i$ such that

$$\| \bar{x}_j - \bar{x}_i \| = d_i = \min_{l \in I - \{i\}} (\| \bar{x}_l - \bar{x}_i \|) \tag{5}$$

Since the theory ([8]) allows only for a global value of σ (and not for a different one for each unit), the average nearest neighbor distance was used:

$$\sigma = \frac{1}{n} \sum_{i \in I} d_i \tag{6}$$

with n the number of examples in the learning set. If the learning set spans uniformly the space where the function is defined, no problems arise. If this is not the case over/under generalization may be expected in this simple scheme.

During the classification task, incomplete feature vectors may be presented to the RBF module. Two strategies were investigated. The first one is probably the simplest: all possible combination of the available data are tested. Its major drawback is that the number of such combinations grows quite rapidly with the number of the available features. The second strategy is directly related to the idea of *characteristic views* of a three dimensional object (see [11]). It can be shown that the space of the possible perspective views of a three dimensional object can be partitioned into subspaces with the following properties:

- all the views in a single subspace are topologically equivalent (their projected edge-structure exhibit a junction line identity);
- every view in a single subspace can be transformed into another view of the same subspace with a linear transformation (in homogeneous coordinates).

An equivalence relation can then be defined and the quotient space considered. The elements of the quotient space are called *characteristic views*. The topological equivalence of the views in the same subspace implies that each view has the same number of features (such as vertices and faces), the reverse being not necessarily true. It is then natural to assign an RBF module to each characteristic view.

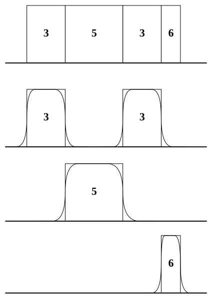

Fig. 2. A schematic view of the *modified characteristic view* approach

Let us focus on Gaussian RBFs. If a sufficiently dense learning set is available, the value returned by the RBF approximation quickly decays with the distance from the learning set. It is reasonable to consider the examples in the learning set as wire frame objects, so that there are no occlusions (somewhat like CAD models). We can then estimate the value of σ on this *virtual* set. The learning set can be subdivided into subsets whose members share the same number of visible (*real*) features. Of course, an equal number of features is necessary but not sufficient to share the same characteristic view. The diagram of Fig. 2 shows a possible scenario and how it can be mapped into different RBF modules, one for each dimensionality (instead of one for each characteristic view) of the learning examples.

The use of a reduced dimensionality implies the use of a modified σ. As σ is directly related to a distance, the following transformation rule was adopted:

$$\sigma_j^2 = \frac{j}{n}\sigma_n^2 \tag{7}$$

where σ_j is the value at dimensionality j. Whenever a feature vector with a dimensionality that does not match any of the available examples is presented to the recognition module, the first strategy can be employed. We must, however, solve the problem of how to compute the distance between points for which some coordinates may not be available. Let us define the metric in the following way:

$$d(\bar{x}_j, \bar{x}_i) = (\bar{x}_j - \bar{x}_i)^T M (\bar{x}_j - \bar{x}_i) \tag{8}$$

where M is a symmetric, positive definite matrix in $\mathcal{M}_{n\times n}$ called the *metric matrix*. If the standard euclidean metric is used and the learning set is complete the following metric can be used ([10]):

$$M_{ij} = g_i \delta_{ij} \tag{9}$$

where δ_{ij} is the Kronecker symbol, $g_i = 1$ if the i-th coordinate is available in the point to be classified and $g_i = 0$ otherwise. A more useful metric can also be defined ([10]) whose diagonal elements take into account the number of missing coordinates. If we choose to preserve the trace of the (diagonal) metric matrix $M \in \mathcal{M}_{n\times n}, \mathrm{Tr}(M) = n$, the elements of the metric matrix are given by:

$$M_{ij} = \frac{n}{\sum_{i=1}^{n} g_i} g_i \delta_{ij} \tag{10}$$

This corresponds to working in a reduced dimensionality RBF module with an effective σ chosen according to the equation 7. The resulting mixed strategy proved to be quite effective, especially with high occlusions percentages. The use of multiple RBF modules with the mixed strategy and a weighted metric provides uniform results, so that the answer of the different modules can be compared to determine the best answer.

To gauge the performance of the RBF scheme, a Nearest Neighbor classification scheme was used on the same data sets. Several additional experiments not reported here have also been performed ([14]).

For each experiment two quantities are shown in the figures:

MIN/MAX: the minimum error made by the module on a wrong clip divided by the maximum error on a right clip. When $MIN/MAX > 1$ the module can avoid false alarm through the introduction of a threshold (see [9]). The average MIN/MAX value over the different modules is reported in the graphs.

RECOGNITION: is computed assigning each clip to the module with the smallest error, ignoring any information from MIN/MAX.

5. Performance analysis

The comparison of "vanilla" RBF and Nearest Neighbor classification seems to favor the latter (see Fig. 3). This can be explained by some recent theoretical results. It can be shown ([5], see also [9]) that under ortographic projections, the view of a given object as defined earlier, spans a 6-dimensional subspace. If the viewer is at a reasonable distance from the object there is only a negligeable difference between perspective and ortographic projection. This imply that the views of each clip almost certainly span non-intersecting 6-manifolds, embedded in the R^{12} representation space. In these experiments, the standard euclidean metrics was used. The use of Gaussians effectively corresponds to setting a volume "around" the 6-manifold. This volume is bigger when the number of examples is smaller (larger inter samples distances) and this can explain the inferior performance of the RBF recognition scheme, which is overgeneralizing. The average MIN/MAX error is better for RBF even if the worst case (worst MIN/MAX figure of the recognition modules) is usually slightly better for NN.

It is interesting to compare the performance of exhaustive matching and multiple modules matching ([14]). Whereas at low occlusion percentages the performance of the two strategies is very similar, the multiple modules approach performs definitely better with a large number of occlusions. The use of a weighted metric increases all the performance measures for both strategies. This improvement is due to the fact that when some features are missing we require a progressively closer match for those we have: this allows the reduction of false alarms and of overgeneralization.

Another possible output representation, which we did not use because it can be proved to yield an inferior performance in this case ([14]), is that of a prototype view ([6]): every view of a given clip is mapped into a particular view of that clip, called its *prototype view* (this mapping is realized through a vector valued function). The clip is then assigned to the module exhibiting the smallest distance from the output of the RBF expansion to its prototype view.

The differences in performance using only positive examples, positive examples with information from negative examples (RBF centers only on positive examples) and positive and negative examples (centers on both positive and negative examples) have been tested using radial Gaussian functions (see Fig.

4). Due to the decay-to-zero of Gaussians negative examples are effective for the vanilla RBF scheme only when the number of examples is small.

6. Hyper Basis Functions technique

The "vanilla" RBF technique provides a basis for comparison with the extensions of the Regularization network technique described by T. Poggio and F. Girosi in [8]. From a more general formulation of the variational problem of regularization they derive the following approximation scheme, instead of equation (1):

$$f^*(\mathbf{x}) = \sum_{\alpha=1}^{n} c_\alpha h(\|\mathbf{x} - \mathbf{t}_\alpha)\|_W^2) + p(\mathbf{x}) \qquad (11)$$

where the parameters $\mathbf{t}_\alpha$, that we call "centers," and the coefficients c_α are unknown, and are in general much fewer than the data points ($n \leq N$). The term $p(\mathbf{x})$ is a polynomial that often can be neglected, though it usually consists of the constant and linear terms. The norm is a *weighted norm*

$$\|(\mathbf{x} - \mathbf{t}_\alpha)\|_W^2 = (\mathbf{x} - \mathbf{t}_\alpha)^T W^T W (\mathbf{x} - \mathbf{t}_\alpha) \qquad (12)$$

where W is an unknown square matrix and the superscript T indicates the transpose. In the simple case of diagonal W the diagonal elements w_i assign a specific weight to each input coordinate, determining in fact the units of measure and the importance of each feature. In this formulation the learning stage is used to estimate not only the coefficients of the RBF expansion, but also the metric (problem dipendent dimensionality reduction) and the position of the centers (optimal examples selection).

This more general optimization problem cannot be solved through matrix inversion and methods like gradient descent ([20]) or random minimization techniques ([13], [12]) must be used (as the possibility of getting trapped in poor local minima is significant we opted for the latter approach).

The possibility of moving the expansion centers, initially located on some of the learning examples, is useful in the presence of noise in the available data and for improving the *representativity* of the center. The adjustable metric is even more important, since it allows to set a more specific dissimilarity measure between new views and centers than euclidean metric.

One of the most interesting results is that in the task of object recognition, and if an approach similar to ours is used, an adjustable metric requires necessarily negative examples. The argument, substantiated by experiments, rests on the trivial observation that if only positive examples were used, it would be possible to obtain an optimal (in terms of quadratic error on the examples) approximation to the characteristic function of an object by mapping every possible input view to the value 1, that is by choosing $\mathbf{W} = 0$. This is obviously a poor choice for the task of object discrimination. The use of negative examples is therefore essential. The added complexity is small since the computational complexity of the RBF expansion depends linearly on the number of examples (for a fixed number of centers). This means that if minimization techniques such as gradient descent or random minimization are used, the increase in the computation time is expected to be linear. In the case of object recognition, negative examples allow to remedy in an elegant way what is the main problem in most learning tasks: the need of a large number of examples. Use of a large number of positive examples would make the whole approach of this paper quite moot: each object would be represented essentially as a look-up table of very many of its views with all the corresponding complexity of storage and acquisition. Our proposal, supported by our experiments, is, instead, to use a small number of positive views and a large number of negative examples, which are easily available if the data basis includes many objects .

The plots of Fig. 5 gives some information on how the generalization proceeds with the number of positive and negative examples available as well as a comparison with the nearest neighbor results on the same data sets. RBF(N) refers to a one center HyperBF expansion with movable coefficient, center and diagonal metric (N is the number of positive examples).

The experimental results allow us to rank the different RBF approaches together with the Nearest Neighbor classification scheme providing a useful gauge. The RBF approaches can be sorted, by increasing performance, in the following way:

1. "vanilla" RBF: use of only positive examples with as many centers as available examples
2. RBF with negative examples: only the positive examples are used as centers but the information from the negative examples is used
3. Nearest Neighbor: the performance is nearly equal to that of RBF with negative examples
4. HyperBF with diagonal metric (*negative* information must be used)
5. HyperBF with complete metric: again negative information must be used.

7. Conclusions

We have applied recently proposed networks for learning from examples (the vanilla RBF version as well as the more powerful HyperBF scheme) to the problem of 3D object recognition. Experimental results on recognition rate have been obtained for wire frame objects (paper clips represented as polylines, hence with no occlusions) and for more realistic objects (paper clips represented as a set of cilinders of varying radii, hence exhibiting different percentages of occlusions). We have extended the RBF technique in order to cope with the problem of feature occlusion through the introduction of a modified euclidean metric. A characteristic view representation has been compared to an exhaustive search for the best match in the case of self-occluded clips. Especially interesting is the use of negative examples which yield some improvement in the vanilla RBF approach and are critically important for the more general HyperBF scheme, while reducing considerably the complexity of the technique in terms of representation and model acquisition. The HyperBF generalization of the basic technique, with the introduction of movable expansion centers and the synthesis of a task dependent metric, proved to be successful in obtaining a compact representation of the objects that appears to work well in the - admittedly still very limited - recognition tasks described here.

References

[1] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981.

[2] D. W. Thompson and J. L. Mundy. Three-dimensional model matching from an unconstrained viewpoint. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 208–220, 1987.

[3] D. G. Lowe. *Perceptual organization and visual recognition.* Kluwer Academic Publishers, 1986.

[4] S. Ullman. Aligning pictorial descriptions: an approach to object recognition. *Cognition*, 32:193–254, 1989.

[5] R. Basri and S. Ullman. Recognition by linear combinations of models. Technical report, The Weizmann Institute of Science, 1989.

[6] T. Poggio and S. Edelman. A network that learns to recognize three-dimensional objects. *Nature*, 343(6225):1–3, 1990.

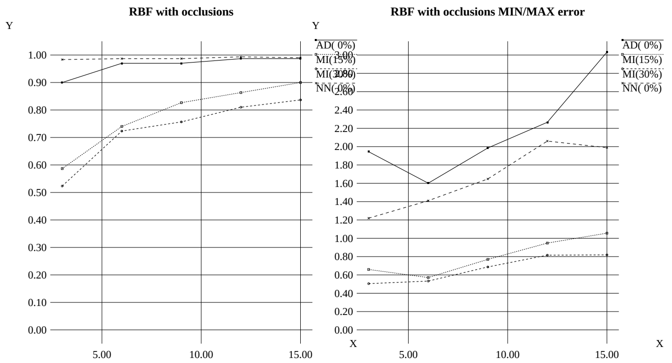

Fig. 3. Comparison between RBF with complete vertex information (AD) and the multiple modules strategy (MI) at different occlusion percentages.

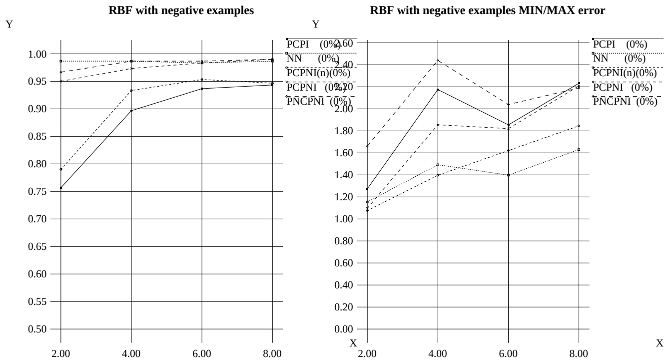

Fig. 4. Recognition rates using information from negative examples: PCPNI uses as centers only the positive examples but takes advantage of the negative examples information (PCPNI(n) uses only positive centers to compute σ); PNCPNI uses as centers all of the positive and negative examples; PCPI does not use negative examples and is provided for comparison.

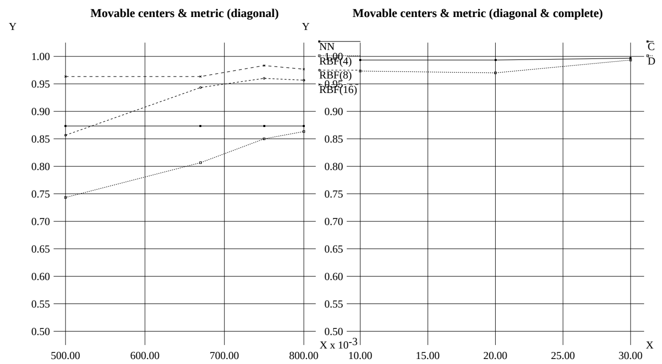

Fig. 5. LEFT: recognition rates using Gaussian HyperBF as a function of the percentage of negative example in the learning set. RIGHT: comparison between complete metric (C) and diagonal metric (D) (both with one movable center). Abscissas represent the number of positive example used. All the positive examples of every other object were used as negative examples.

[7] T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.

[8] T. Poggio and F. Girosi. A theory of networks for approximation and learning. Technical Report A.I. Memo No. 1140, Massachusetts Institute of Technology, 1989.

[9] S. Edelman and T. Poggio. Bringing the grandmother back into the picture: a memory-based view of object recognition. Technical Report A.I. Memo No. 1181, Massachusetts Institute of Technology, 1990.

[10] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice Hall, 1988.

[11] H. Freeman and I. Chakravarty. The use of characteristic views in the recognition of three dimensional objects. In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition In Practice*, pages 277–288. North-Holland, 1980.

[12] G. Tecchiolli and R. Brunelli. On random minimization of quadratic forms. Technical Report in prep., I.R.S.T, 1990.

[13] B. Caprile and F. Girosi. A nondeterministic minimization algorithm. Technical Report A.I. Memo No. 1254, Massachusetts Institute of Technology, 1990.

[14] R. Brunelli and T. Poggio. Use of rbf in real object recognition. Technical Report 9011-09, I.R.S.T, 1990.

[15] T. Poggio and F. Girosi. Networks for approximation and learning. In *Proc. of the IEEE, Vol. 78*, pages 1481–1497, 1990.

[16] D. P. Huttenlocher and S. Ullman. In *Proc. 1st Int. Conf. Computer Vision*, pages 102–111, Washington, D.C., 1987.

[17] C. A. Micchelli. Interpolation of scattered data: Distance matrices and conditionally positive definite functions. *Constr. Approx*, 2:11–22, 1986.

[18] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice Hall, Englewood Cliffs, NJ, 1982.

[19] S. Suzuki and K. Abe. Binary picture thinning by an iterative parallel two-subcycle operation. *Pattern Recognition*, 20(3):297–307, 1987.

[20] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1988.

[21] S. Edelman and D. Weinshall. A self-organizing multiple-view representation of 3d objects. Technical Report A.I. Memo No. 1146, Massachusetts Institute of Technology, 1989.