MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY
and
CENTER FOR BIOLOGICAL AND COMPUTATIONAL LEARNING
DEPARTMENT OF BRAIN AND COGNITIVE SCIENCES

A.I. Memo No. 1615 September, 1997
C.B.C.L Paper No. 154

Visual recognition and categorization on the basis of similarities to multiple class prototypes

Shimon Edelman and Sharon Duvdevani-Bar

This publication can be retrieved by anonymous ftp to publications.ai.mit.edu.
The pathname for this publication is: ai-publications/1500-1999/AIM-1615.ps.Z

Abstract

One of the difficulties of object recognition stems from the need to overcome the variability in object appearance caused by factors such as illumination and pose. The influence of these factors can be countered by learning to interpolate between stored views of the target object, taken under representative combinations of viewing conditions. Difficulties of another kind arise in daily life situations that require categorization, rather than recognition, of objects. We show that, although categorization cannot rely on interpolation between stored examples, knowledge of several representative members, or prototypes, of each of the categories of interest can still provide the necessary computational substrate for the categorization of new instances. The resulting representational scheme based on similarities to prototypes is computationally viable, and is readily mapped onto the mechanisms of biological vision revealed by recent psychophysical and physiological studies.

Some of the results described here appeared in a preliminary form in the Philosophical Transactions of the Royal Society, **352B**(1358), pp.1191-1202 (1997). This report describes research done at the Center for Biological and Computational Learning in the Department of Brain and Cognitive Sciences at the Massachusetts Institute of Technology. S. Duvdevani-Bar is at the Department of Applied Mathematics & Computer Science, Weizmann Institute of Science, Rehovot 76100, Israel.

1 Introduction

To be able to recognize objects, a visual system must combine the capacity for internal representation and for the storage of object traces with the ability to compare these against the incoming visual stimuli, namely, images of objects. The appearance of an object is determined not only by its shape and surface properties, but also by its disposition with respect to the observer and the illumination sources, by the optical properties of the intervening medium and the imaging system, and by the presence and location of other objects in the scene (Ullman, 1996). Thus, to detect that two images belong, in fact, to the same three-dimensional object, the visual system must overcome the influence of a number of factors that affect the way objects look.

The choice of approach to the separation of the intrinsic shape of an object from the extrinsic factors affecting its appearance depends on the nature of the task faced by the system. One of these tasks, which may be properly called *recognition* (knowing a previously seen object as such), appears now to require little more than storing information concerning earlier encounters with the object, as suggested by the success of view-based recognition algorithms developed in computer vision in early 1990's (Poggio and Edelman, 1990; Ullman and Basri, 1991; Breuel, 1992; Tomasi and Kanade, 1992). In this paper, we show that it is surprisingly easy to extend such a memory-based strategy to deal with *categorization*, a task that requires the system to make sense of novel shapes. Thus, familiarity with a relatively small selection of objects can be used as a foundation for processing (i.e., representing and categorizing) other objects, never seen before.

The theory of representation on which the present approach is based calls for describing objects in terms of their similarities to a relatively small number of reference shapes (Edelman, 1995b; Edelman, 1997b). The theoretical underpinnings of this idea are discussed elsewhere (Edelman and Duvdevani-Bar, 1997); here, we demonstrate its viability on a variety of objects and object classes, and discuss the implications of its successful implementation for understanding object representation and categorization in biological vision.

1.1 Visual recognition

If the appearance of visual objects were immutable and unaffected by any extrinsic factors, recognition would amount to simple comparison by template matching, a technique in which two patterns are regarded as the same if they can be brought into one to one correspondence. As things stand, the effects of the extrinsic factors must be mitigated to ensure that the comparison is valid. Theories of recognition, therefore, tend to have two parts: one concentrating on the form of the internal representation into which images of objects are cast, and the other on the details of the comparison process.

A model of recognition that is particularly well-suited to the constraints imposed by a biological implementation has been described in (Poggio and Edelman, 1990). This model relies on the observation that the views of a rigid object undergoing transformation such as rotation in depth reside in a smooth low-dimensional manifold embedded in the space of coordinates of points attached to the object (Ullman and Basri, 1991; Jacobs, 1996); furthermore, the properties of smoothness and low dimensionality of this *view space* manifold are likely to be preserved in whatever measurement space is used by the front-end of the visual system. The operational consequence of this observation is that a new view of an object may be recognized by interpolation among its selected stored views, which together represent the object. A criterion that indicates the quality of the interpolation can be formed by comparing the stimulus view to the stored views, by passing the ensuing proximity values through a Gaussian nonlinearity, and by computing a weighted sum of the results (this amounts to a basis-function interpolation of the view manifold, as described in section 3.1). The outcome of this computation is an estimate of the measurement-space distance between the point that encodes the stimulus and the view manifold. If a sufficient number of views is available to define that manifold, this distance can be made arbitrarily independent of the pose of the object, one of the extrinsic factors that affect the appearance of object views. The influence of the other extrinsic factors (e.g., illumination) can be minimized in a similar manner, by storing examples that span the additional dimensions of the view manifold, corresponding to the additional degrees of freedom of the process of image formation.

In the recognition scenario, the tacit assumption is that the stimulus image is either totally unfamiliar, or, in fact, corresponds to one of the objects known to the system. A sensible generic decision strategy under this assumption is *nearest-neighbor* (Cover and Hart, 1967), which assigns to the stimulus the label of the object that matches it optimally (modulo the influence of the extrinsic factors, and, possibly, measurement noise). In the view-interpolation scheme, the decision can be based on the value of the distance-to-the-manifold criterion that reflects the quality of the interpolation (a low value signifies an unfamiliar object). As we argue next, this approach, being an instance of the generic nearest-neighbor strategy, addresses only a small part of the problem of visual object processing.

1.2 Visual categorization

Because it assumes that variability in object appearance is mainly due to factors such as illumination and pose, the standard approach to recognition calls for a comparison between the intrinsic shape of the viewed object (separated from the influence of the extrinsic factors) and the stored representation of that shape. According to this view, a good representation is one that makes explicit the intrinsic shape of an object in great detail and with high fidelity.

A reflection on the nature of everyday recognition tasks prompts one to question the validity of this view of representation. In a normal *categorization* situation (Rosch, 1978; Smith, 1990), human observers are expected to *ignore* much of the shape details (Price and Humphreys, 1989). Barring special (albeit behaviorally important) cases such as face recognition, entry-level

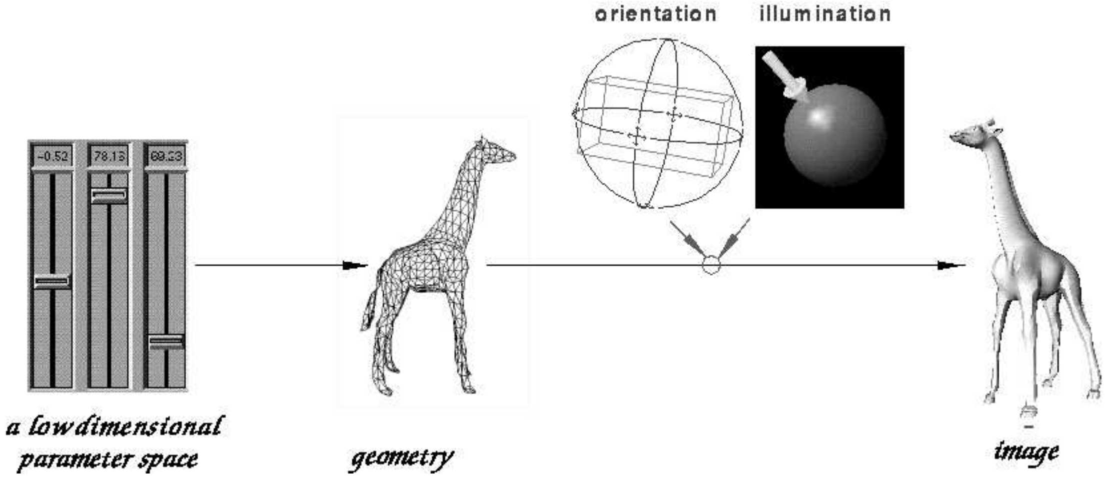


Figure 1: The process of image formation. A family of shapes (say, 4-legged animal-like objects) can be defined parametrically, using a small number of variables (Edelman and Duvdevani-Bar, 1997), illustrated symbolically on the left by the three "sliders" that control the values of the shape variables. These, in turn, determine the geometry of the object, e.g., the locations of the vertices of a triangular mesh that approximates the object's shape. Finally, intrinsic and extrinsic factors (geometry and viewing conditions) together determine the appearance of the object.

(Jolicoeur et al., 1984) names of objects correspond to categories rather to individuals, and it is the category of the object that the visual system is required to determine. Thus, the observer is confronted with potential variation in the intrinsic shape of an object, because objects called by the same name do not, generally, have exactly the same shape. This variability in the shape (and not merely in the appearance) of objects must be adequately represented, so that it can be treated properly at the categorization stage.

Different gradations of shape variation call for different kinds of action on the part of the visual system. On the one hand, moderately novel objects can be handled by the same mechanism that processes familiar ones, insofar as such objects constitute variations on familiar themes. Specifically, the nearest-neighbor strategy around which the generic recognition mechanism is built can be allowed to handle shape variation that does not create ambiguous situations in which two categories vie for the ownership of the current stimulus. On the other hand, if the stimulus image belongs to a radically novel object — e.g., one that is nearly equidistant, in the similarity space defined by the representational system, to two or more familiar objects, or very distant from any such object — a nearest-neighbor decision no longer makes sense, and should be abandoned in favor of a better procedure. Such a procedure, suitable for representing both familiar and novel shapes, is described in the next section.

2 The shape space

To be able to treat familiar and novel shapes uniformly within the same representational framework, it is useful to describe shapes as points in a common parameter space. A common parameterization is especially straightforward for shapes that are sampled at a preset resolution, then defined by the coordinates of the sample points (cf. Figure 1). For instance, a family of shapes each of which is a "cloud" of k points spans a $3k$-dimensional *shape space* (Kendall, 1984); moving the k points around in 3D (or, equivalently, moving around the single point in the $3k$-dimensional shape space) amounts to changing one shape into another.

By defining similarity between shapes via a distance function in the shape space, clusters of points are made to correspond to classes of shapes (i.e., sets of shapes whose members are more similar to each other than to members of other sets). To categorize a (possibly novel) shape, then, one must first find the corresponding point in the shape space, then determine its location with respect to the familiar shape clusters. Note that while a novel shape may fall in between the clusters, it will in any case possess a well-defined representation. This representation may be then acted upon, e.g., by committing it to memory, or by using it as a seed for establishing a new cluster.

2.1 The high-dimensional measurement space

Obviously, a visual system has no direct access to whatever shape space in which the geometry of distal objects may be defined (in fact, the notion of a unique geometrical shape space does not even make sense: the same physical object can be described quantitatively in many different ways). The useful and intuitive notion of a space in which each point corresponds to some shape can, however, be put to work by introducing an intermediary concept: *measurement space*.

A system that carries out a large number of measurements on a visual stimulus effectively maps that stimulus into a point in a high-dimensional space; the diversity and the large number of independent measurements increase the likelihood that any change in the geometry of the distal objects ends up represented at least in some of the dimensions of the measurement space. Indeed, in primate vision, the dimensionality of the space presented by the eye to the brain is roughly one million – the same as the number of fibers in each optic nerve.

Most of this high-dimensional space is empty: a randomly chosen combination of pixel values in an image is extremely unlikely to form a picture of a coherent object. The locus of the measurement-space points that do represent images of coherent objects depends on all the factors that participate in image formation, both intrinsic (the shapes of objects) and extrinsic (e.g., their pose), which together define the *proximal* shape space. Note that smoothly changing the shape of the imaged object causes the corresponding point to ascribe a manifold in the measurement space. The dimensionality of this manifold depends on the number of degrees of freedom of the shape changes; for example, simple morphing of one shape into another produces a one-dimensional manifold (a curve). Likewise, rotating the object in depth (a transformation with two degrees of freedom) gives rise to a two-dimensional manifold which we call the view space of the object. It turns out that the proximal shape space, produced by the joint effects of deformation and transformation, can be safely considered a locally smooth low-dimensional manifold embedded in the measurement space (Edelman and Duvdevani-Bar, 1997).

2.2 Dimensionality reduction and the proximal shape space

In the above formulation, the categorization problem becomes equivalent to determining the location of the measurement-space representation of the stimulus within the proximal shape space. Our approach to this problem is inspired by the observation that the location of a point can be precisely defined by specifying its distance to some prominent reference points, or *landmarks* (Edelman and Duvdevani-Bar, 1997). Because distance here is meant to capture difference in shape (i.e., the amount of deformation), its estimation must exclude (1) components of measurement-space distance that are orthogonal to the shape space, as well as (2) components of shape transformation such as rotation. As we shall see, a convenient computational mechanism for distance estimation that satisfies these two requirements is a module tuned to a particular shape, that is, designed to respond selectively to that shape, irrespective of its transformation. A few such modules, tuned to different reference shapes, effectively reduce the dimensionality of the representation from that of the measurement space to a small number, equal to the number of modules (Figure 2). In the next section, we describe a system for shape categorization based on a particular implementation of this approach, which we call the Chorus of Prototypes (Edelman, 1995b); its relevance as a model of shape processing in biological vision is discussed in section 5.

3 The implementation

A module tuned to a particular shape will fulfill the first of the two requirements stated above – ignoring the irrelevant components of the measurement-space distance – if it is trained to discriminate among objects all of which belong to the desired shape space. Such a training imparts to the module the knowledge of the relevant measurement-space directions, by making it concentrate on the features that help discriminate between the objects. To fulfill the second requirement – insensitivity to shape transformations – the module must be trained to respond equally to different views of the object to which it is tuned. A trainable computational mechanism capable of meeting these two requirements is a radial basis function (RBF) interpolation module.

3.1 The RBF module

When stated in terms of an input-output relationship, our goal is to build a module that would output a nonzero constant for any view of a certain target object, and zero for any view of all the other objects in the training set. Because only a few target views are usually available for training, the problem is to *interpolate* the view space of the target object, given some examples of its members. With basis function interpolation (Broomhead and Lowe, 1988), this problem can be solved by a distributed network, whose structure can be learned from examples (Poggio and Girosi, 1990).

According to this method, the interpolating function is constructed out of a superposition of basis functions, whose shape reflects the prior knowledge concerning the change in the output as one moves away from the data point. In the absence of evidence to the contrary, all directions of movement are considered equivalent, making it reasonable to assume that the basis function is radial (that is, it depends only on the distance between the actual input and the original data point, which serves as its center). The resulting scheme is known as radial basis function (RBF) interpolation. Once the basis functions have been placed, the output of the interpolation module for any test point is computed by taking a weighted sum of the values of all the basis functions at that point.

An application of RBF interpolation to object recognition has been described in (Poggio and Edelman, 1990); the RBF model was subsequently used to replicate a number of central characteristics of the process of recognition in human vision (Bülthoff and Edelman, 1992). In its simple version, one basis function is used for (the measurement-space representation of) each familiar view. The appropriate weight for each basis is then computed by an algorithm that involves matrix inversion (a closed-form solution exists for this case). This completes the process of training the RBF network. To determine whether a test view belongs to the object on which the network has been trained, this view (that is, its measurement-space representation) is compared to each of the training views. This step yields a set of distances between the test view and the training views that serve as the centers of the basis functions. In the next step, the values of the basis functions are combined linearly to determine the output of the network (see Figure 3, inset, and appendix A).

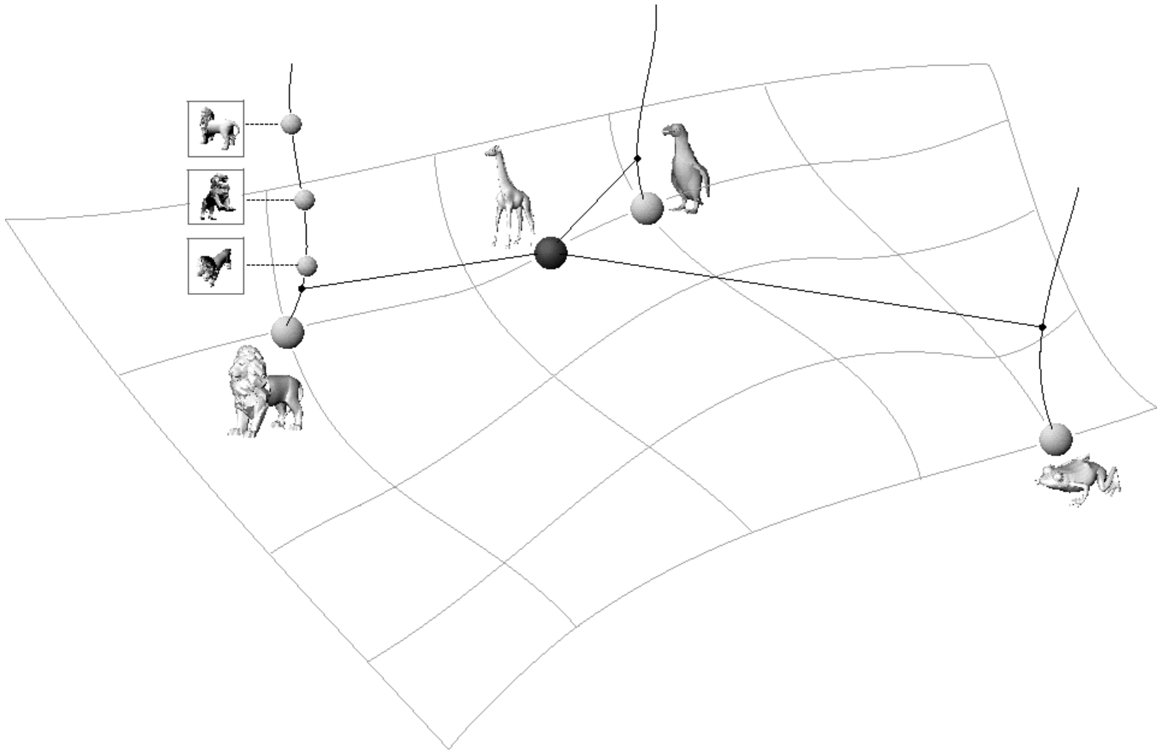


Figure 2: A schematic illustration of the shape-space manifold defined by a Chorus of three active modules (`lion`, `penguin`, `frog`). Each of the three reference-shape modules is trained to ignore the viewpoint-related factors (the view space dimension, spanned by views that are shown explicitly for `lion`), and is thus made to respond to shape-related differences between the stimulus (here, the `giraffe`) and its "preferred" shape. The actual dimensionality of the space spanned by the outputs of the modules (Edelman and Intrator, 1997) can be lower than its nominal dimensionality (equal to the number of modules); here the space is shown as a two-dimensional manifold.

3.2 Multi-classifier network design

A multi-classifier network is constructed by combining several single-shape modules, each tuned to a different shape class. The multi-classifier network is trained according to the algorithm described in appendix C.1. The response properties of such a network are illustrated in Figure 16, which shows the activity of several RBF modules for a number of views of each of the objects on which they had been trained. As expected, each module's response is the strongest for views of its preferred shape, and is weaker for views of the other shapes. Significantly, the response is rarely very weak; this feature contributes to the distributed nature of the representation formed by an ensemble of modules, by making several modules active for most stimuli.[1]

It has been hypothesized (Edelman et al., 1996) that the ensemble of responses produced by a collection of object-specific modules can serve as a substrate for carrying out classification of the stimulus at superordinate, basic, or subordinate levels of categorization (Rosch et al., 1976; Rosch, 1978), depending on the manner in which the response vector is processed. In the next section we describe a series of computational experiments that examine the representational capabilities of a multi-classifier network in a range of tasks.

4 Experimental results

In all our computational experiments we used three-dimensional object geometry data available as a part of a commercial database that contains several hundreds of shapes. Ten reference objects were chosen at random from the database, to serve as the prototypes for the multi-classifier network implementation of the Chorus scheme (see Figure 5).

To focus on the problem of shape-based recognition, objects were rendered under the Lambertian shading assumption, using a simulated point light source situated at the camera, a uniform gray surface color, and no texture. Each object was presented to the system separately, on a white background, at the center of a 256×256 window; the maximal dimensions of the 3D bounding boxes of the objects were normalized to a standard size (about one half of the size of the window). Thus, the problems of figure-ground segmentation and of transla-

[1]Note that much more information concerning the shape of the stimulus is contained in the entire pattern of activities that it induces over the ensemble of the reference-object modules, compared to the information in the identity of the strongest-responding module (Edelman et al., 1992). Typical object recognition systems in computer vision, which involve a Winner Take All decision, opt for the latter, impoverished, representation of the stimulus.

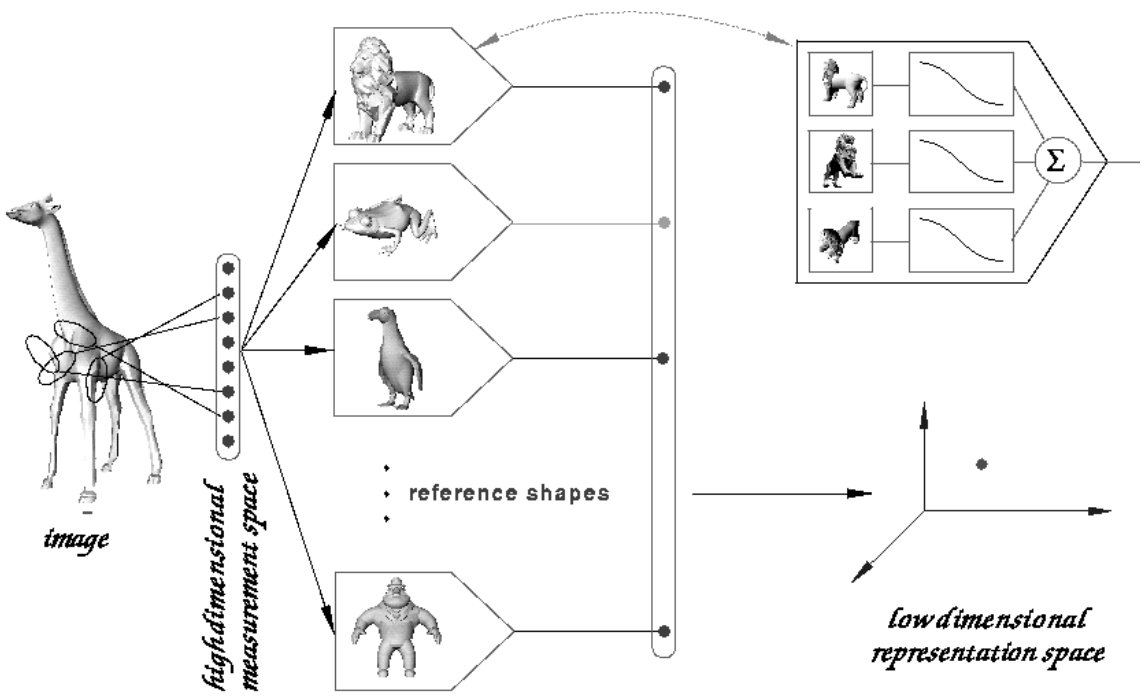


Figure 3: The Chorus scheme (section 3). The stimulus is first projected into a high-dimensional measurement space, spanned by a bank of receptive fields. Second, it is represented by its similarities to reference shapes. In this illustration, only three modules respond significantly, spanning a shape space that is nominally three-dimensional (in the vicinity of the measurement-space locus of giraffe images). The *inset* shows the structure of each module. Each of a small number of training views, $\mathbf{v}_t$, serves as the center of a Gaussian basis function $\mathcal{G}(\mathbf{a}, \mathbf{b}; \sigma) = exp\left(\|\mathbf{a} - \mathbf{b}\|^2/\sigma^2\right)$; the response of the module to an input vector $\mathbf{x}$ is computed as $y = \sum_t w_t \mathcal{G}(\mathbf{x}; \mathbf{v}_t)$. The weights w_t and the spread parameter σ are learned as described in (Poggio and Girosi, 1990). It is important to realize that the above approach, which amounts to an interpolation of the view space of the training object using the radial basis function (RBF) method, is not the only one applicable to the present problem. Other approaches, such as interpolation using the multilayer perceptron architecture, may be advantageous, e.g., when the measurement space is "crowded," as in face discrimination (Edelman and Intrator, 1997).

tion and scale invariance were effectively excluded from consideration.

The performance of the resulting 10-module Chorus system was assessed in three different tasks: (1) *identification* of novel views of the ten objects on which the system had been trained, (2) *categorization* of 43 novel objects belonging to categories of which at least one exemplar was available in the training set, and (3) *discrimination* among 20 novel objects, chosen at random from the database.

4.1 Identification of novel views of familiar objects

The ability of the system to generalize identification to novel views was tested on the ten reference objects, for each of which we had trained a dedicated RBF module. We experimented with three different identification algorithms, whose performance was evaluated on a set of 169 views, taken around the canonical orientation specific for each object (Palmer et al., 1981). The test views ranged over $\pm 60^\circ$ in azimuth and elevation, at 10° increments.

4.1.1 Identification results

We first computed the performance of each of the ten RBF modules using individually determined thresholds. For each module, the threshold was set to the mean activity on trained views[2] less one standard deviation. The performance of each of the ten modules on its training object is summarized in Table 1. As one can see, the residual error rates were about 10%, a figure that can probably be improved if a more powerful architecture or a more extensive learning procedure are used. The generalization error rate (defined as the mean of the miss and the false alarm rates, taken over all ten reference objects) for the individual-threshold algorithm was 7%.

We next considered the Winner-Take-All (WTA) algorithm, according to which the outcome of the identification step is the label of the module that gives the strongest response to the current stimulus (in Table 4, appendix D, entries for modules that responded on the average the strongest are marked by bold typeface). The error rate of the WTA method was 10%.

[2]Only about a tenth of the 169 views, determined by canonical vector quantization (see appendix B.1), had been used in training the modules.

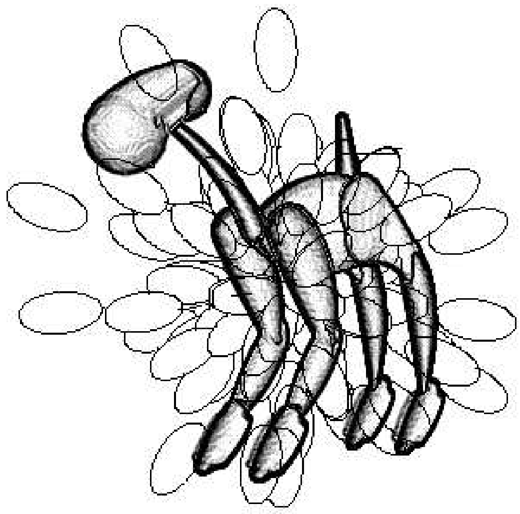


Figure 4: An image of a 3D object, overlayed by the outlines of the receptive fields (RFs) used to map object views into a high-dimensional measurement space (see section 2.1 and appendix B). The system described here involved 200 radially elongated Gaussian RFs; only some of them are drawn in this figure.

	cow1	cat	Al	gene	tuna	Lrov	Niss	F16	fly	TRex
miss rate	0.11	0.14	0.02	0.01	0.13	0.04	0.03	0.10	0.16	0.05
false alarm rate	0.08	0.11	0.07	0.02	0.11	0.05	0.04	0.12	0.12	0.03

Table 1: Individual shape-specific module performance. The table shows the miss and the false alarm rates of modules trained on the objects shown in Figure 5. The generalization error rate (defined as the mean of the miss and the false alarm rates) was 7%.

Finally, we trained a second-level RBF module to map the 10-element vector of the outputs of the reference-object modules into another 10-dimensional vector only one of whose elements (corresponding to the actual identity of the input) was allowed to assume a nonzero value of 1; the other elements were set to 0 (Edelman et al., 1992). This approach takes advantage of the distributed representation of the stimulus by postponing the Winner Take All decision until after the second-level module has taken into account the similarities of the stimulus to *all* reference objects. Indeed, the WTA algorithm applied to the second-level RBF output resulted in an error rate of 6%.

4.1.2 Lessons from the identification experiments

The purpose of the first round of experiments was to ensure that the system of reference-object modules could be trained to identify novel views of those objects. The satisfactory performance of the RBF modules, which did generalize to novel views of the training objects, allowed us to proceed to test the entire system in a number of representation scenarios involving novel shapes, as described below. We note that one cannot expect the performance on novel objects to be better than that on the familiar ones. Thus, the figure obtained in the present section — about 10% error rate — sets a bound on the performance in the other tasks. To improve that, one may attempt to employ an alternative learning mechanism (as suggested above), in conjunction with a better image transduction stage, instead of the 200 Gaussian RFs we used here.

4.2 Categorization of novel object views

Our second experiment tested the ability of the Chorus scheme to categorize "moderately" novel stimuli, each of which belonged to one of the categories present in the original training set of ten objects. To that end, we used the 43 test objects shown in Figure 6. To visualize the utility of representation by similarity to the training objects, we used multidimensional scaling (Shepard, 1980) to embed the 10-dimensional layout of points corresponding to various views of the test objects into a two-dimensional space (Figure 7). An examination of the resulting plot revealed two satisfying properties. First, views of various objects clustered by object identity (and not, for instance, by pose, as in patterns derived by multidimensional scaling from distances measured in the original pixel space). Second, in Figure 7 views of the `QUADRUPEDS`, the `AIRPLANES` and the `CARS` categories all form distinct "super-clusters."

To assess the quality of this representation numerically, we used it to support object categorization. A number of categorization procedures were employed at

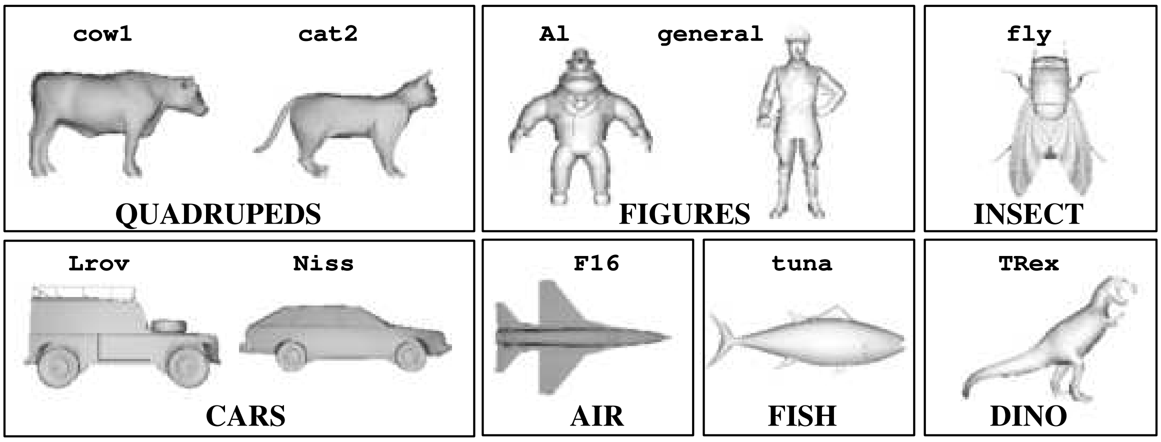


Figure 5: The ten training objects used as reference shapes in the computational experiments described in the text, organized by object categories. The objects were chosen at random from a collection available from Viewpoint Datalabs, Inc. (http://www.viewpoint.com/).

this stage. In every case, the performance of the 10-dimensional Chorus-based representation was compared to that of the original multidimensional receptive-field (RF) measurement space (see Figure 4; we shall return to discuss this comparison later on).

The various categorization procedures we used were tested on the same set of 169 views per object as before. First, we assigned a category label to each of the ten training objects (for instance, `cow` and `cat` were both labeled as `QUADRUPEDS`). Second, we represented each test view as a 10-element vector of RBF-module responses. Third, we employed a categorization procedure to determine the category label of the test view. Each view that was attributed to an incorrect category by the categorization procedure was counted as an error.

The category labels we used are the same as the labels given to the various groups of objects in Figure 5. Note that a certain leeway exists in the assignment of the labels. Normally, these are determined jointly by a number of factors, of which shape similarity is but one. For example, a `fish` and a `jet aircraft` are likely to be judged as different categories; nevertheless, if the shape alone is to serve as the basis for the estimation of their similarity, these categories may coalesce. We tested this assumption in an independent psychophysical experiment (Duvdevani-Bar, 1997), in which human subjects were required to judge similarity among the same shapes used in the present study, on the basis of shape cues only. Similarity scores[3] obtained in this experiments revealed a clustering of object shapes in which the `fly` belonged to the `FIGURES` category, and `AIR`craft were interspersed within the `FISH` category.

A careful examination of the confusion tables produced by the different categorization methods we describe below revealed precisely these two phenomena as the major sources of miscategorization errors. First, the `fly` classifier turned out to be highly sensitive to the members of the `FIGURES` category. Second, the `tuna` module was in general more responsive to `AIR`craft than the `F16` module (the sole representative of `AIR`craft among the reference objects). To quantify the effects of this ambiguity in the definition of category labels on performance, we compared three different sets of labels for the reference objects. The first set of category labels is the one shown in Figure 5. The second set differs from the first one in that it labels the `fly` as a `FIGURE`; in the third set, the `tuna` and the `F16` have the same category label.

4.2.1 Winner-Take-All (WTA)

According to the WTA algorithm, the label of the module that produces the strongest response to the novel stimulus determines its category membership. We note that the WTA method is incompatible with the central tenet of the Chorus approach — that of distributed representation. To be informative, a representation based on similarities to reference objects requires that more than one module respond to any given stimulus. A system trained with this requirement in mind is expected to thwart the WTA method by having different modules compete for a given stimulus, especially when the latter does not quite fit into any of the familiar object categories. Indeed, in this experiment the WTA algorithm yielded a high misclassification rate of 45% over the 43 test objects for the first set of category labels. Adding a second-stage RBF module trained as described in section 4.1 reduced this figure to 30%. When the second and the third set of category labels were used, misclassification rate decreased to 32%, and 25%, respectively. Carrying out the WTA algorithm in the second-stage RBF space reduced both those figures to 23%.

4.2.2 k-NN using multiple views

We next examined another simple categorization method, based on the k Nearest Neighbor (k-NN) principle (Duda and Hart, 1973). The categorization module

[3] Score data were gathered using the tree construction method (Fillenbaum and Rapoport, 1979), and were submitted to multidimensional scaling analysis (SAS procedure MDS, 1989) to establish a spatial representation of the different shapes.

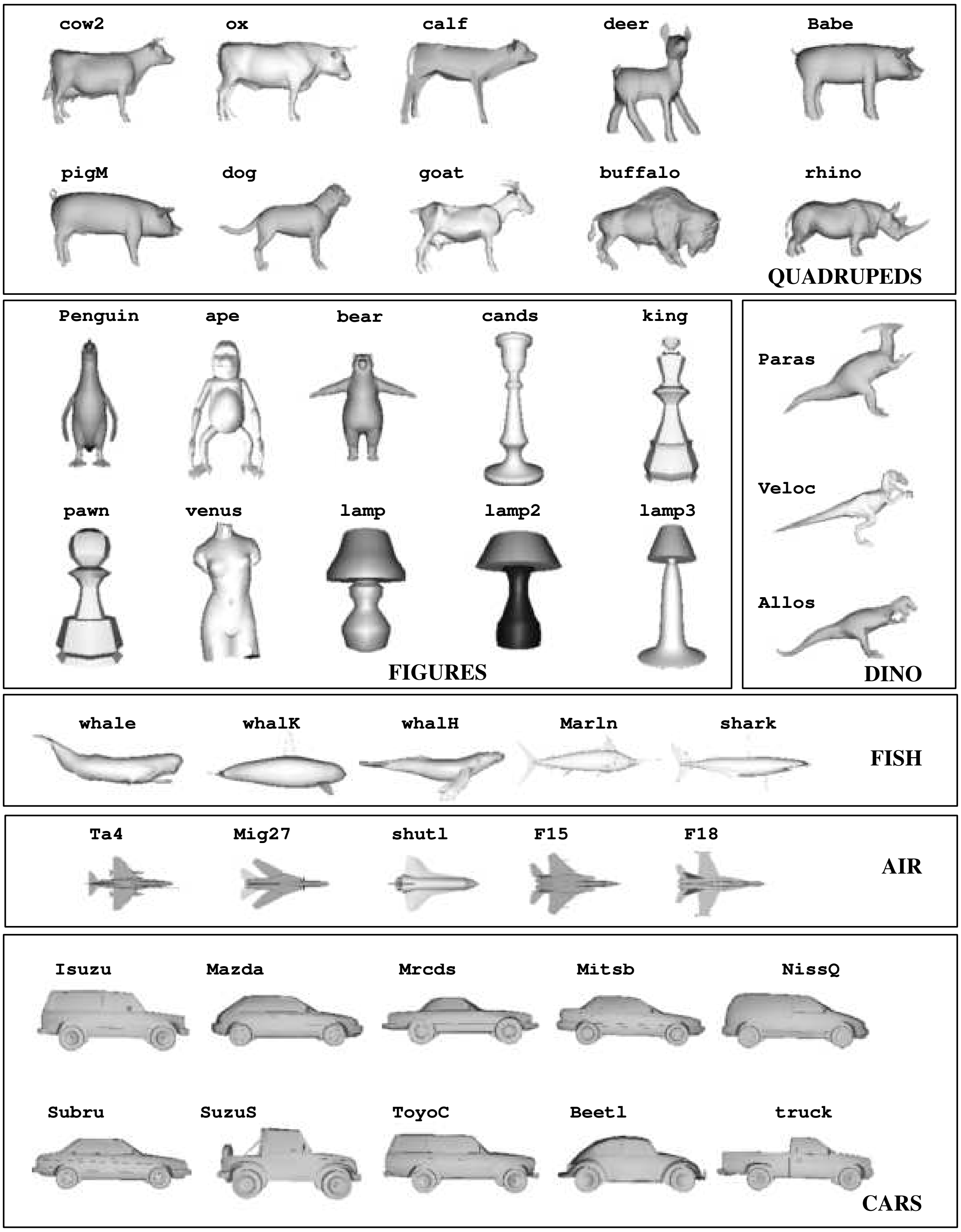


Figure 6: The 43 novel objects used to test the categorization ability of the model (see section 4.2); objects are grouped by shape category.

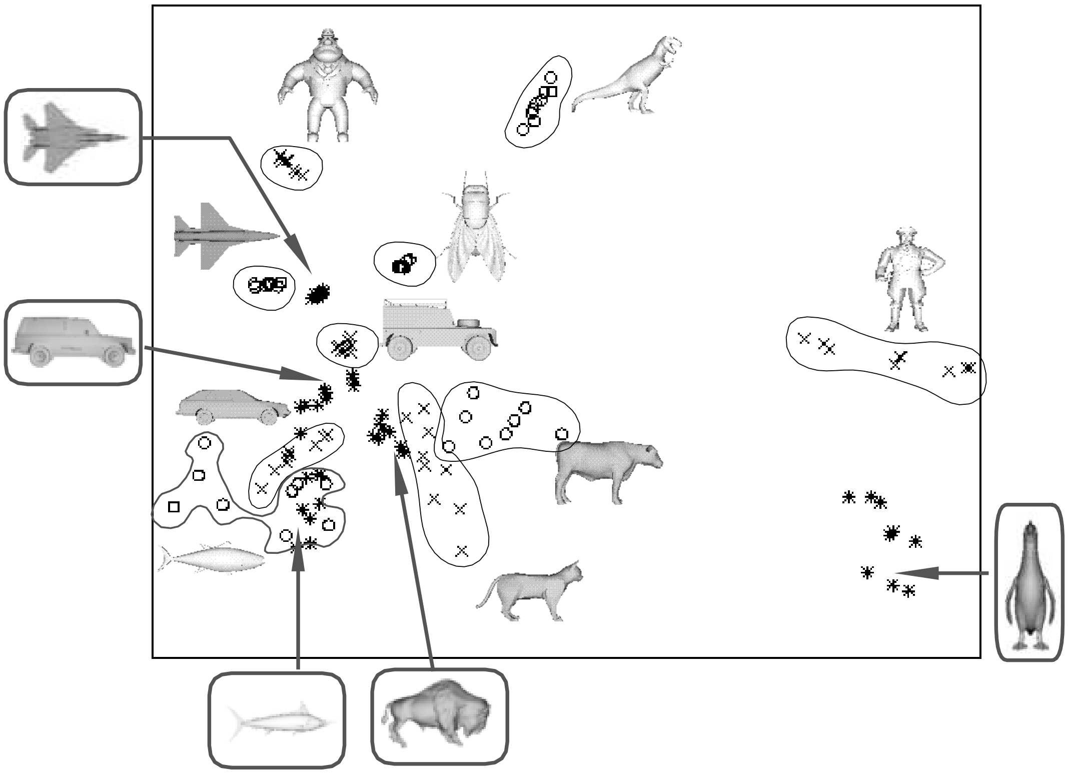


Figure 7: A 2D plot of the 10-dimensional shape space spanned by the outputs of the RBF modules; multidimensional scaling (MDS) was used to render the 10D space in 2D, while preserving as much as possible distances in the original space (Shepard, 1980). Each point corresponds to a test view of one of the objects; nine views of each of the ten training and five novel objects (`buffalo`, `penguin`, `marlin`, `Isuzu`, `F15`, marked by $*$'s). Note that views belonging to the same object tend to cluster (part of the residual spread of each cluster can be attributed to the constraint, imposed by MDS, of fitting the two dimensions of the viewpoint variation *and* the dimensions of the shape variation into the same 2D space of the plot). Note also that clusters corresponding to similar objects (e.g., the `QUADRUPEDS`) are near each other. The icons of the objects appear near the corresponding view clusters; those of five novel objects are drawn in cartouche.

was made to store N views of each reference object, each represented as a point in the 10-dimensional space of module outputs ($10N$ views altogether were stored). The category of a test view was then determined by polling the k reference views that turned out to be the closest to the test view in the 10D space. The label of the majority of those k views was assigned to the test view.

The performance of this method for the third set of category labels is summarized in Figure 8, which shows the categorization rates for different values of k and N, averaged over the 43 test objects. Note that the misclassification error rate decreases with the number of views considered, possibly because the relative amount of reliable information available in the neighborhood of the test view increases. In contrast, the tendency to err increases with k. The mean misclassification rate for this set of labels was 29% (41% and 31% for the first and second cases, respectively). In comparison, when the 200-dimensional measurement space was used to represent the individual views, the mean error rate was 37%, 34%, and 32% for the first, second and third sets of category labels, respectively.

4.2.3 1-NN using centers of view clusters

A variation on the above method is to use clusters of views of the reference objects, rather than individual views. If the clusters are tight, their centroids approximate them well. Accordingly, we used the centroid of the set of training views of each object (cast into the 10D space) as the representative member of that object's cluster. Categorization followed the Nearest Neighbor principle, which, in line with the notation of the preceding section, may be called the 1-NN algorithm. This procedure resulted in misclassification rates of 20%, 17%, and 15% for the three different sets of category labels. The 1-NN procedure showed a clear benefit of the 10-dimensional RBF-module representation over the 200-dimensional measurement space, where the same procedure yielded misclassification rates of 30%, 25%, and 23%, for the three sets of category labels.

4.2.4 k-NN to the training views

The previous method assumed that clusters are well-represented by their means, which is not necessarily true in practice. Likewise, the assumption that an unlimited number of views of the training objects is available for use in the scheme of section 4.2.2 is not always justified. The use of all and only those views that were actually employed in the training of the 10 RBF modules circumvents both these problems. Thus, the last categorization method we tested involved the k-NN algorithm along with the training views specific to each of the RBF modules. At the first level of the RBF representation space, this method yielded mean misclassification rate of 23%, 16% and 14% for the three sets of category labels; average is taken over values of k ranging from 1 to 9. In the measurement space, the misclassification rates were slightly higher; on average over the same values of k, misclassification rates for the three category label sets were 23%, 22% and 20%. Tables 6 (in appendix D) and 2 give the detailed errors obtained for the third set of category labels, for $k = 3$. Note how the definition of category labels of the reference objects affects the resulting misclassification rate.

4.2.5 Lessons from the categorization experiments

The pattern of the performance of the various algorithms we tested in the categorization tasks conforms to the expectations. Specifically, the RBF representation was better than the "raw" 200-dimensional measurement space. Although the latter outcome was not uniform (as apparent in the nearly identical performance of the RBF and the measurement spaces in some conditions), it was quite consistent under conditions that we consider more realistic (e.g., when view-cluster centers, or the actual training views were used in the representation; see sections 4.2.3 and 4.2.4), and for the more appropriate definitions of the categorization task (i.e., for the second and third sets of category labels).

Despite those encouraging results, the performance of the system in the categorization experiments (about 80%) falls short by 10−15% of the human performance in comparable circumstances. We list possible explanations of this shortcoming in the general discussion section.

4.3 Discrimination among object views

Our third experiment tested the ability of the Chorus scheme to represent 20 novel objects (shown in Figure 9), picked at random from the database, and to support their discrimination from one another. The tests involved the same arrangement of 169 views per object as before. The representation of the test objects is described in Table 5, which shows the activation of the ten reference-shape RBF modules produced by each of the test objects.

4.3.1 Discrimination results

It is instructive to consider the patterns of similarities revealed in this distributed 10-dimensional representation of the test objects. For instance, the `giraffe` turns out to be similar to the two quadrupeds present in the training set (`cow` and `cat`), as well as to the dinosaur (`TRex`), for obvious reasons (it is also similar to the `tuna` and to the `fly`, for reasons which are less obvious, but immaterial: both these reference shapes are similar to most test objects, which makes their contribution to the representation uninformative). Thus, in the spirit of Figure 2, the `giraffe` can be represented by the vector [1.87 1.93 1.72] of similarities to the three reference objects which turn out to be informative in this discrimination context (`cow`, `cat`, `TRex`).

As in Figure 7, the model clustered views by object identity, and grouped view clusters by similarity between the corresponding objects. In a quantitative estimate of this performance, we used the k-NN algorithm, as explained in section 4.2.2, with labels correspond to object identity rather than to object category. The k-NN procedure that relied on proximities to the 169 views of each of the reference objects yielded a mean error rate (averaged over values of k ranging from 1 to 9) of 5% over the 169 test views of the 20 novel objects. When only 25 views spanning the range of $\pm 20°$ around the canonical orientation of each test object were considered, the mean

Category labeling	QUAD	FIGS	FISH	AIR	CARS	DINO
Set I	0.08	0.34	0.14	0.50	0.11	0.33
Set II	0.08	0.10	0.14	0.50	0.11	0.33
Set III	0.08	0.10	0.14	0.28	0.11	0.33

Table 2: The individual errors for each category of test objects (see Table 6 for details). Note how the error rates decrease for the test objects of the `FIGURES` category in the second case, and for the test objects of the `AIR` category in the third case.

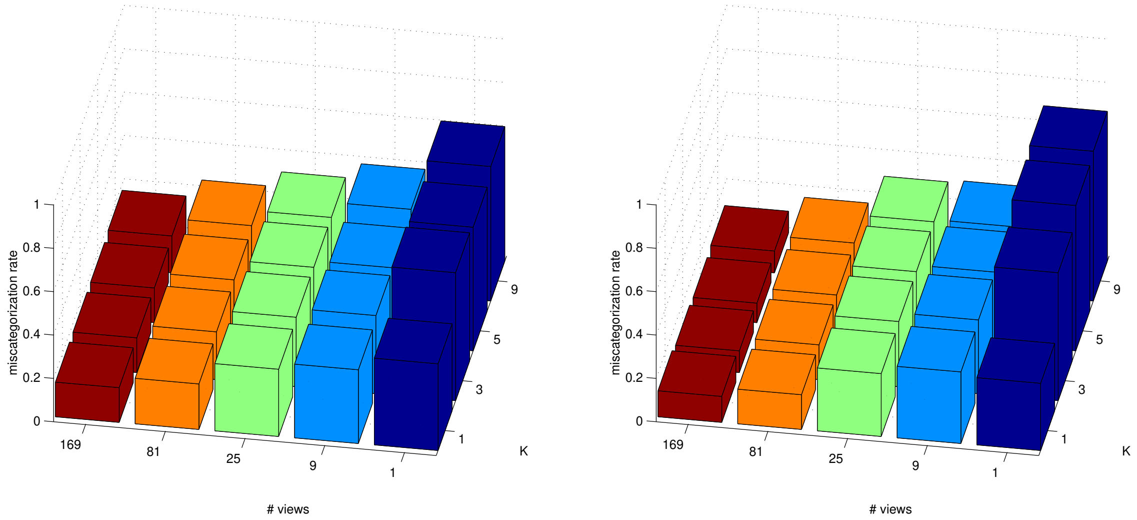


Figure 8: The performance of the k-NN procedure described in section 4.2.2 for the third set of category labels, plotted vs. k and N. The plots show the misclassification rate for the 43 test objects shown in Figure 6. *Left:* errors using the measurement-space representation; the mean misclassification error is 32%. *Right:* the same, for the RBF-module representation space; the mean misclassification error is 29%.

error rate dropped to 1.5%. This improvement may be attributed in part to the exclusion of non-representative views, e.g., the head-on view of the `manatee`, which is easily confused with the top view of the `lamp`. In the RF-representation case, same experiment yielded error rate of 1% with respect to the 169 views, whereas no error occurred when 25 views of all 20 objects were considered.

When the same procedure was carried out for the 43 test objects of Figure 6, error rate was on the average higher, because these objects resemble each other more closely. The mean error rate (averaged over values of k ranging from 1 to 9) for the 169 test views of the 43 objects was 15% in the RBF space and 7% in the RF-representation space.

4.3.2 Lessons from the discrimination experiments

When objects are highly dissimilar from one another, discrimination (which requires that the objects be represented with the least possible confusion) is relatively easy. In that case, the measurement space representation is effective enough. To see that, one may compare the discrimination results obtained with the measurement-space representation of the set of 20 highly distinct novel objects of Figure 9 to the results obtained with the same method on the measurement-space representation of the 43 objects (Figure 6) used before. The advantage of the measurement-space representation over the RBF space in some discrimination tasks stems from the higher dimensionality and hence higher informativeness of the former. This high dimensionality is, however, a liability rather than an asset in generalization and other categorization tasks, an observation that is supported by our data.

To quantify the ability of the model to reduce the dimensionality of the measurement space, we estimated its performance with a varying number of reference objects, holding the size of the test set fixed. In addition, we quantified the extent of dimensionality reduction that could be afforded under the constraint of a specific preset discrimination error. Figure 10, left, shows the discrimination error rate obtained with the 3-NN method described in section 4.2.2 (using 25 views per test object), plotted against the number of reference and test objects (see also Table 7 in appendix D). Figure 10, right, shows the number of reference objects required to perform the discrimination task (using the 3-NN method on 25 views per test object) with an error rate less than 10%, for a varying number of test objects. To the extent that it could be tested with the available data, the scaling of the model's performance with the number of test objects seems to be satisfactory.

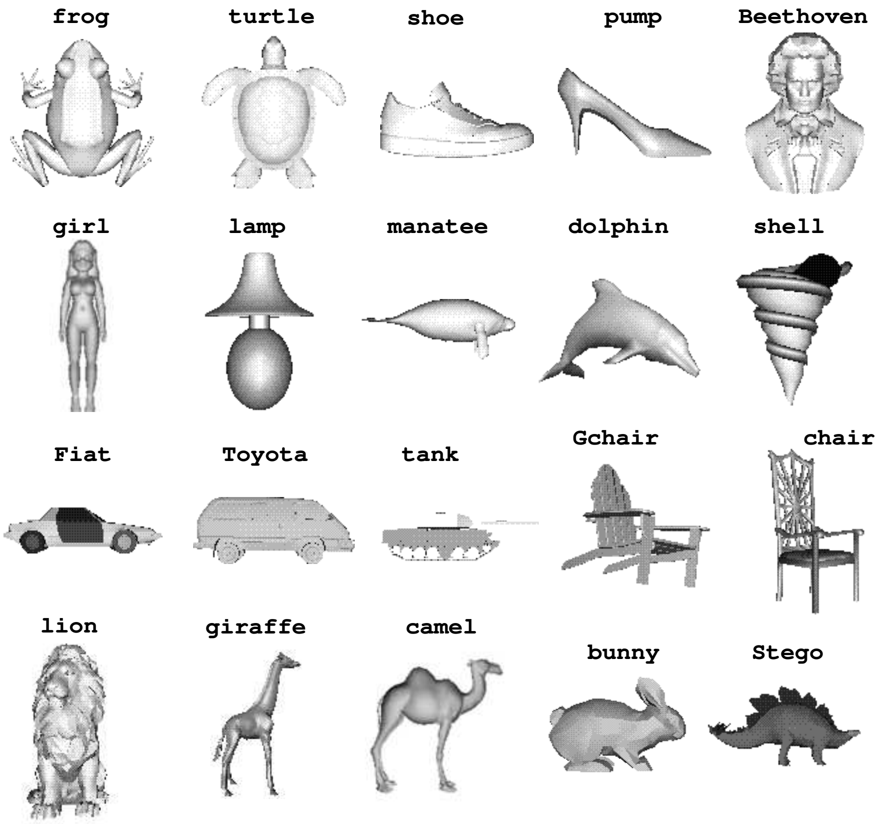
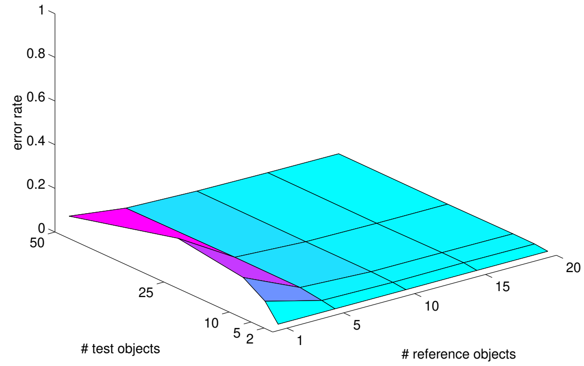


Figure 9: The 20 novel objects, picked at random from the object database, which we used to test the representational abilities of the model (see section 4.3).

Repr. Space	Category labeling	Method: WTA 1st	WTA 2nd	k-NN M	1-NN	k-NN C
RBF	Set I	45	30	41	20	23
	Set II	32	25	31	17	16
	Set III	23	23	29	15	14
RF	Set I			37	30	23
	Set II			34	25	22
	Set III			32	23	20

Table 3: A summary of misclassification error rates exhibited by the various methods of section 4.2, for the three sets of category labels, using both the 200-dimensional measurement space and the 10-dimensional RBF representation space. The error rate improved with each categorization method we introduced. The Winner-Take-All (WTA) of section 4.2.1 produced the highest error, which was reduced when a second-stage RBF module was added. The k-NN method of section 4.2.2, using multiple views around the test view, produced similar error rates, which were significantly improved by using centers of view clusters (1-NN) (see section 4.2.3), or when the k-NN method involving the training views was used (section 4.2.4). For the last three methods, the error obtained in the RF measurement space was higher that the corresponding error obtained in the RBF space. Note that under all methods, the errors improved when the second and the third sets of category labels were used.



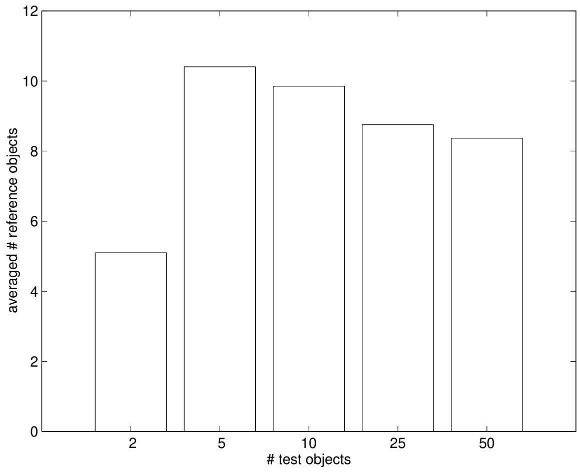


Figure 10: *Left:* the mean discrimination error rate plotted against the representation dimensionality (the number of reference objects) and the size of the test set (the number of test objects). The means were computed over 10 random choices of reference and test objects. See Table 7 in appendix D for performance figures. *Right:* the dimensionality of the representation (the number of reference objects) required to perform discrimination with an error rate of 10% or less, for a varying number of test objects. The data for this plot were obtained by repeating the task of discriminating among the views of N_t test objects represented by the activities of N_p reference objects 2500 times; this corresponded to 10 independent choices of N_t test objects out of a set of 50 test objects (five values of N_t were tested: 2,5,10,25,50), and to 10 random selections of $N_p = 1, 5, 10, 15, 20$ out of the 20 available reference objects.

5 Discussion

5.1 Implications for theories of visual representation

In computer vision, one may discern three main theoretical approaches to object representation: pictorial representations, structural descriptions, and feature spaces (Ullman, 1989). According to the first approach, objects are represented by the same kind of geometric information one finds in a picture: *coordinates* of primitive elements, which, in turn, may be as simple as intensity values of pixels in an image (Lowe, 1987; Ullman, 1989; Poggio and Edelman, 1990; Ullman and Basri, 1991; Breuel, 1992; Tomasi and Kanade, 1992; Vetter et al., 1997). Because of the effects of factors extrinsic to shape, this mode of representation can be used for recognition only if it is accompanied by a method for normalizing the appearance of objects (Ullman, 1989) or, more generally, for separating the effects of pose from the effects of shape (Ullman and Basri, 1991; Tomasi and Kanade, 1992).

It is not easy to adapt the pictorial approach to carry out categorization rather than recognition. One reason for that is the excessive amount of detail in pictures: much of the information in a snapshot of an object is unnecessary for categorization, as attested by the ability of human observers to classify line drawings of common shapes (Biederman and Ju, 1988; Price and Humphreys, 1989). Although a metric over images that would downplay within-category differences may be defined in some domains, such as classification of stylized "clip art" drawings (Ullman, 1996, p.173), attempts to classify pictorially represented 3D objects (vehicles) met with only a limited success (Shapira and Ullman, 1991).

We believe that extension of alignment-like approaches from recognition to categorization is problematic for a deeper reason than mere excess of information in images of objects. Note that both stages in the process of recognition by alignment (normalization and comparison; see Ullman, 1989) are geared towards pairing the stimulus with a *single* stored representation (which may be the average of several actual objects, as in Basri's 1996 algorithm). As we pointed out in the introduction, this strategy, designed to culminate in a winner-take-all decision, is inherently incompatible with the need to represent radically novel objects.

The ability to deal with novel objects has been considered so far the prerogative of structural approaches to representation (Marr and Nishihara, 1978; Biederman, 1987). The structural approach employs a small number of generic primitives (such as the thirty-odd geons postulated by Biederman), along with spatial relationships defined over sets of primitives, to represent a very large variety of shapes. The classification problem here is addressed by assigning objects that have the same structural description to the same category.

In principle, even completely novel shapes can be given a structural description, because the extraction of primitives from images and the determination of spatial relationships is supposed to proceed in a purely bottom-up, or image-driven fashion. In practice, however, both these steps proved so far impossible to automate. State of the art recognition systems in computer vision tend to ignore the challenge posed by the problems of categorization and of representation of novel objects (Murase and Nayar, 1995), or treat categorization as a kind of imprecise recognition (Basri, 1996).

In contrast to all these approaches, the Chorus model is designed to treat both familiar and novel objects equivalently, as points in a shape space spanned by similarities to a handful of reference objects. According to Ullman's (1989) taxonomy, this makes it an instance of the feature-based approach, the features being similarities to entire objects. The minimalistic implementation of Chorus described in the preceding sections achieved recognition and generalization performance comparable to that of the state of the art computer vision systems (Murase and Nayar, 1995; Mel, 1997; Schiele and Crowley, 1996), despite relying only on shape cues where other systems use shape and color or texture or both. Furthermore, this performance was achieved with a low-dimensional representation (ten nominal dimensions), whereas the other systems typically employ about a hundred dimensions; for a discussion of the importance of low dimensionality in this context, see (Edelman and Intrator, 1997). Finally, our model also exhibited significant capabilities for shape-based categorization and for useful representation of novel objects; it is reasonable to assume that its performance in these tasks can be improved, if more lessons from biological vision are incorporated into the system.

5.2 Implications for understanding object representation in primate vision

The architecture of Chorus reflects our belief that a good way to achieve progress in computer vision is to follow examples set by biological vision. Each of the building blocks of Chorus, as well as its general layout, can be readily interpreted in terms of well-established properties of the functional architecture of the primate visual system (Poggio, 1990; Poggio and Hurlbert, 1994). The basic mechanism in the implementation of this scheme is a *receptive field* — probably the most ubiquitous functional abstraction of the physiologist's tuned unit, widely used in theories of biological information processing (Edelman, 1997a). The receptive fields at the front end of Chorus are intended to parallel those found in the initial stages of the primate visual pathway.[4] Furthermore, an RBF module of the kind used in the subsequent stage of Chorus can be seen also as a receptive field, tuned both to a certain location in the visual field (defined by the extent of the front-end receptive fields) and to a certain location in the shape space (corresponding to the shape of the object on which the module has been trained).

Functional counterparts both of individual components (basis functions) of RBF modules and of entire modules have been found in a recent electrophysiological study of the inferotemporal (IT) cortex in awake monkeys (Logothetis et al., 1995). The former correspond to cells tuned to particular views of objects familiar to the animal; the latter — to cells that respond nearly equally to a wide range of views of the same object. It is easy to imagine how an ensemble of cells of the latter kind, each tuned to a different reference object, can span an internal shape space, after the manner suggested above.

While a direct test of this conjecture awaits experimental confirmation, indirect evidence suggests that a mechanism not unlike the Chorus of Prototypes is deployed in the IT cortex. This evidence is provided by the work of K. Tanaka and his collaborators, who studied object representation in the cortex of anaesthetized monkeys (Tanaka, 1992; Tanaka, 1996). These studies revealed cells tuned to a variety of simple shapes, arranged so that units responding to similar shapes were clustered in columns running perpendicular to the cortical surface; the set of stimuli that proved effective depended to some extent on the monkey's prior visual experience. If further experimentation reveals that a given object consistently activates a certain possibly disconnected subset of the columns, and if that pattern of activation smoothly changes in response to a continuous change in the shape or the orientation (Wang et al., 1996) of the stimulus, the principle of representation of similarity that serves as the basis of Chorus would be implicated also as the principle behind shape representation in the cortex.

The results of several recent psychophysical studies of object representation in primates support the above conjecture. In each of a series of experiments, which involved subjective judgment of shape similarity and delayed matching to sample, human subjects (Edelman, 1995a; Cutzu and Edelman, 1996) and monkeys (Sugihara et al., 1997) have been confronted with several classes of computer-rendered 3D animal-like shapes, arranged in a complex pattern in a common parameter space (cf. Shepard & Cermak, 1973). In each experiment, processing of the subject data by multidimensional scaling (used to embed points corresponding to the stimuli into a 2D space for the purpose of visualization) invariably revealed the low-dimensional parametric structure of the set of stimuli. In other words, the proximal shape space internalized by the subjects formed a faithful replica of the distal shape space structure imposed on the stimuli. Furthermore, this recovery was reproduced by a Chorus-like model, trained on a subset of the stimuli and subsequently exposed to the same test images shown to the subjects. As we argue elsewhere, these findings may help understand the general issue of cognitive representation, and, in particular, the manner in which representation can conform, or be faithful, to its object (Edelman and Duvdevani-Bar, 1997; Edelman, 1997b); their full integration will require a coordinated effort in the fields of behavioral physiology, psychophysics, and computational modeling.

5.3 Summary

We have described a computational model of shape-based recognition and categorization, which encodes stimuli by their similarities to a number of reference shapes, themselves represented by specially trained dedicated modules. The performance of the model (see Table 3) suggests that this principle may allow for efficient representation, and, in most cases, correct categorization, of shapes never before encountered by the observer — a goal which we consider of greater importance than

[4] Admittedly, the 200 elongated-Gaussian RFs used in our present simulations are too crude to serve as a model even of the primary visual cortex. A better preprocessing stage (e.g., a simulation of the complex-cell system described in (Edelman et al., 1997)) should be tested in conjunction with the Chorus scheme.

mere recognition of previously seen objects, and which so far has eluded the designers of computer vision systems.

The most severe limitations of the present model are (1) the lack of tolerance to image-plane translation and scaling of the stimulus, (2) the lack of a principled way of dealing with occlusion and interference among neighboring objects in a scene, and (3) the lack of explicit representation of object structure (a shortcoming it shares with many other feature-based schemes). Whereas it may be possible to treat translation and scaling effectively without abandoning the present approach (Vetter et al., 1995; Riesenhuber and Poggio, 1998), its extension to scenes and to the explicit representation of structure must await future research.

Acknowledgments We thank M. Dill, N. Intrator, T. Poggio and P. Sinha for helpful suggestions concerning an earlier version of this manuscript.

References

Adini, Y., Moses, Y., , and Ullman, S. (1997). Face recognition: the problem of compensating for illumination changes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. in press.

Basri, R. (1996). Recognition by prototypes. *International Journal of Computer Vision*, 19(147-168).

Baxter, J. (1996). The canonical distortion measure for vector quantization and approximation. *Unpublished manuscript*.

Biederman, I. (1987). Recognition by components: a theory of human image understanding. *Psychol. Review*, 94:115–147.

Biederman, I. and Ju, G. (1988). Surface versus edge-based determinants of visual recognition. *Cognitive Psychology*, 20:38–64.

Breuel, T. M. (1992). *Geometric Aspects of Visual Object Recognition*. PhD thesis, MIT.

Broomhead, D. S. and Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355.

Bülthoff, H. H. and Edelman, S. (1992). Psychophysical support for a 2-D view interpolation theory of object recognition. *Proceedings of the National Academy of Science*, 89:60–64.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Trans. on Information Theory*, IT-13:21–27.

Cutzu, F. and Edelman, S. (1996). Faithful representation of similarities among three-dimensional shapes in human vision. *Proceedings of the National Academy of Science*, 93:12046–12050.

Duda, R. O. and Hart, P. E. (1973). *Pattern classification and scene analysis*. Wiley, New York.

Duvdevani-Bar, S. (1997). *Similarity to Prototypes in 3D Shape Representation*. PhD thesis, Weizmann Institute of Science.

Duvdevani-Bar, S. and Edelman, S. (1995). On similarity to prototypes in 3D object representation. CS-TR 95-11, Weizmann Institute of Science.

Edelman, S. (1995a). Representation of similarity in 3D object discrimination. *Neural Computation*, 7:407–422.

Edelman, S. (1995b). Representation, Similarity, and the Chorus of Prototypes. *Minds and Machines*, 5:45–68.

Edelman, S. (1997a). Receptive fields for vision: from hyperacuity to object recognition. In Watt, R., editor, *Vision*. MIT Press, Cambridge, MA. in press.

Edelman, S. (1997b). Representation is representation of similarity. Behavioral and Brain Sciences, to appear.

Edelman, S., Cutzu, F., and Duvdevani-Bar, S. (1996). Similarity to reference shapes as a basis for shape representation. In Cottrell, G. W., editor, *Proceedings of 18th Annual Conf. of the Cognitive Science Society*, pages 260–265, San Diego, CA.

Edelman, S. and Duvdevani-Bar, S. (1997). Similarity, connectionism, and the problem of representation in vision. *Neural Computation*, 9:701–720.

Edelman, S. and Intrator, N. (1997). Learning as extraction of low-dimensional representations. In Medin, D., Goldstone, R., and Schyns, P., editors, *Mechanisms of Perceptual Learning*, pages 353–380. Academic Press.

Edelman, S., Intrator, N., and Poggio, T. (1997). Complex cells and object recognition. in preparation.

Edelman, S., Reisfeld, D., and Yeshurun, Y. (1992). Learning to recognize faces from examples. In Sandini, G., editor, *Proc. 2nd European Conf. on Computer Vision, Lecture Notes in Computer Science*, volume 588, pages 787–791. Springer Verlag.

Fillenbaum, S. and Rapoport, A. (1979). *Structures in the subjective lexicon*. Academic Press, New York.

Gersho, A. and Gray, R. M. (1992). *Vector quantization and signal compression*. Kluwer Academic Publishers, Boston.

Jacobs, D. W. (1996). The space requirements of indexing under perspective projections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:330–333.

Jolicoeur, P., Gluck, M., and Kosslyn, S. M. (1984). Pictures and names: making the connection. *Cognitive Psychology*, 16:243–275.

Kanatani, K. (1990). *Group-theoretical methods in image understanding*. Springer, Berlin.

Kendall, D. G. (1984). Shape manifolds, Procrustean metrics and complex projective spaces. *Bull. Lond. Math. Soc.*, 16:81–121.

Lando, M. and Edelman, S. (1995). Receptive field spaces and class-based generalization from a single view in face recognition. *Network*, 6:551–576.

Linde, Y., Buzo, A., and Gray, R. (1980). An algorithm for vector quantizer design. *IEEE Transactions on Communications*, COM-28:84–95.

Logothetis, N. K., Pauls, J., and Poggio, T. (1995). Shape recognition in the inferior temporal cortex of monkeys. *Current Biology*, 5:552–563.

Lowe, D. G. (1987). Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proc. 5th Berkeley Symposium*, 1:281–297.

Marr, D. and Nishihara, H. K. (1978). Representation and recognition of the spatial organization of three dimensional structure. *Proceedings of the Royal Society of London B*, 200:269–294.

Mel, B. (1997). SEEMORE: Combining color, shape, and texture histogramming in a neurally-inspired approach to visual object recognition. *Neural Computation*, 9:777–804.

Moody, J. and Darken, C. (1989). Fast learning in networks of locally tuned processing units. *Neural Computation*, 1:281–289.

Murase, H. and Nayar, S. (1995). Visual learning and recognition of 3D objects from appearance. *International Journal of Computer Vision*, 14:5–24.

Palmer, S. E., Rosch, E., and Chase, P. (1981). Canonical perspective and the perception of objects. In Long, J. and Baddeley, A., editors, *Attention and Performance IX*, pages 135–151. Erlbaum, Hillsdale, NJ.

Poggio, T. (1990). A theory of how the brain might work. *Cold Spring Harbor Symposia on Quantitative Biology*, LV:899–910.

Poggio, T. and Edelman, S. (1990). A network that learns to recognize three-dimensional objects. *Nature*, 343:263–266.

Poggio, T. and Girosi, F. (1989). A theory of networks for approximation and learning. A.I. Memo No. 1140, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

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

Poggio, T. and Hurlbert, A. (1994). Observations on cortical mechanisms for object recognition and learning. In Koch, C. and Davis, J., editors, *Large Scale Neuronal Theories of the Brain*, pages 153–182. MIT Press, Cambridge, MA.

Price, C. J. and Humphreys, G. W. (1989). The effects of surface detail on object categorization and naming. *Quarterly J. Exp. Psych. A*, 41:797–828.

Riesenhuber, M. and Poggio, T. (1998). Just one view: Invariances in inferotemporal cell tuning. In M. I. Jordan, M. J. K. and Solla, S. A., editors, *Advances in Neural Information Processing*, volume 10, pages –. MIT Press. in press.

Rosch, E. (1978). Principles of categorization. In Rosch, E. and Lloyd, B., editors, *Cognition and Categorization*, pages 27–48. Erlbaum, Hillsdale, NJ.

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., and Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychology*, 8:382–439.

Sas (1989). *SAS/STAT User's Guide, Version 6*. SAS Institute Inc., Cary, NC.

Schiele, B. and Crowley, J. L. (1996). Object recognition using multidimensional receptive field histograms. In Buxton, B. and Cipolla, R., editors, *Proc. ECCV'96*, volume 1 of *Lecture Notes in Computer Science*, pages 610–619, Berlin. Springer.

Shapira, Y. and Ullman, S. (1991). A pictorial approach to object classification. In *Proceedings IJCAI*, pages 1257–1263.

Shepard, R. N. (1980). Multidimensional scaling, tree-fitting, and clustering. *Science*, 210:390–397.

Shepard, R. N. and Cermak, G. W. (1973). Perceptual-cognitive explorations of a toroidal set of free-form stimuli. *Cognitive Psychology*, 4:351–377.

Smith, E. E. (1990). Categorization. In Osherson, D. N. and Smith, E. E., editors, *An invitation to cognitive science: Thinking*, volume 2, pages 33–53. MIT Press, Cambridge, MA.

Sugihara, T., Edelman, S., and Tanaka, K. (1997). Representation of objective similarity among three-dimensional shapes in the monkey. *Biological Cybernetics*. in press.

Tanaka, K. (1992). Inferotemporal cortex and higher visual functions. *Current Opinion in Neurobiology*, 2:502–505.

Tanaka, K. (1996). Inferotemporal cortex and object vision. *Annual Review of Neuroscience*, 19:109–139.

Tomasi, C. and Kanade, T. (1992). Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9:137–154.

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

Ullman, S. (1996). *High level vision*. MIT Press, Cambridge, MA.

Ullman, S. and Basri, R. (1991). Recognition by linear combinations of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:992–1005.

Vetter, T., Hurlbert, A., and Poggio, T. (1995). View-based models of 3d object recognition: Invariance to imaging transformations. *Cerebral Cortex*, 5:261–269.

Vetter, T., Jones, M. J., and Poggio, T. (1997). A bootstrapping algorithm for learning linear models of object classes. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 40–46, Puerto Rico.

Wang, G., Tanaka, K., and Tanifuji, M. (1996). Optical imaging of functional organization in the monkey inferotemporal cortex. *Science*, 272:1665–1668.

Weiss, Y. and Edelman, S. (1995). Representation of similarity as a goal of early visual processing. *Network*, 6:19–41.

A Theoretical aspects of the design of a shape-tuned module

In this section, we study the theoretical underpinnings of the ability of an RBF module to overcome the variability induced by pose changes. Specifically, we show that once an RBF module is trained on a collection of object views, its response to views that differ from its centers (the training examples) in a small displacement *along* the view space spanned by the examples is always higher than its response to views that are orthogonal to or directed away from this view space.

A.1 The infinitesimal displacement case

Assume the view space of a specific object shape can be sampled, and consider the sketch given in Figure 11, illustrating the following notation:

- $\mathbf{x}_1$ is a training view, $\mathbf{x}_i$ — another, arbitrary, training view, $i = 1, \ldots, k$.
- $\Delta\mathbf{x}$ — a unit vector, $(\Delta\mathbf{x})^T \Delta\mathbf{x} = 1$.
- $t > 0$, a parameter controlling the extent of the displacement in the direction of $\Delta\mathbf{x}$.

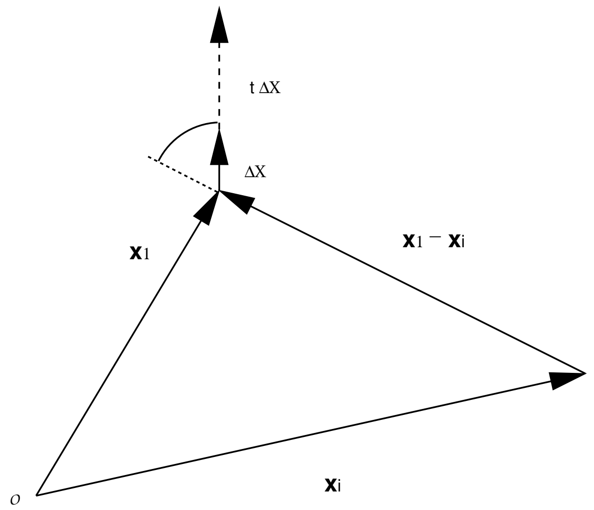


Figure 11: An illustration of the basic notations used in the text; $\mathbf{x}_1$, $\mathbf{x}_i$ are training views of a specific object shape, $i = 1, \ldots, k$. $t\Delta\mathbf{x}$ is a vector representing a displacement from the view space spanned by the training vectors. The angle between $t\Delta\mathbf{x}$ and $\mathbf{x}_1 - \mathbf{x}_i$ indicates the direction of displacement. When *all* such angles are sharp, the displacement is away from the view space, whereas when there is at least one such angle that is obtuse, the displacement is towards one of the $\mathbf{x}_i$'s, and therefore towards the view space.

Assume further that we train a (Gaussian) RBF network on a set of pairs $\{\mathbf{x}_i, y_i\}_{i=1}^k$, for $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^k$, a set of that object views, and a simple target $\mathbf{y} = \{y_i = 1\}_{i=1}^k$. For an input vector $\mathbf{x}$, the corresponding $RBF(\mathbf{x})$ activity is given by:

$$\begin{aligned} RBF(\mathbf{x}) &= \sum_{i=1}^{k} c_i \; G(\|\mathbf{x} - \mathbf{x}_i\|) \\ &= \sum_{i=1}^{k} c_i \; e^{-[(\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}-\mathbf{x}_i)]^2/\sigma^2}. \end{aligned} \tag{1}$$

Let $\mathbf{A} = (a_i)$, $\mathbf{B} = (b_j)$, define $\mathbf{G}(\mathbf{A}; \mathbf{B})$ to be a matrix whose entry (i,j) is the Gaussian $e^{\frac{-\|a_i - b_j\|^2}{\sigma^2}}$. Training in its simplest form means solving the equation

$$y = \mathbf{G}(\mathbf{x}; \mathbf{X}) \cdot \mathbf{c},$$

for the value of $\mathbf{c}$. The solution is:

$$\mathbf{c} = \mathbf{G}^{+}(\mathbf{X}; \mathbf{X}) \cdot \mathbf{y}, \tag{2}$$

where $^+$ denotes the (pseudo) inverse of $\mathbf{G}$.

Thus, equation (1) takes the form

$$RBF(\mathbf{x}) = \mathbf{G}(\mathbf{x};\mathbf{X}) \cdot \mathbf{G}^{+}(\mathbf{X};\mathbf{X}) \cdot \mathbf{y}. \tag{3}$$

Upon successful training, $RBF(\mathbf{x}_1) = 1 - \epsilon,\ \ \epsilon \ll 1$. We now compute the change in RBF behavior resulting from an infinitesimal displacement from a training vector $\mathbf{x}_1$, in an arbitrary direction.

$$\begin{aligned}
&\frac{\partial RBF(\mathbf{x}+t\Delta\mathbf{x})}{\partial t}\,\Big|_{\substack{\mathbf{x}=\mathbf{x}_1\\ t>0,\ t\to 0}} = \\
&\frac{\partial}{\partial t}\left[\sum_{i=1}^{k} c_i\ e^{-[(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)]^2/\sigma^2}\right] = \\
&\sum_{i=1}^{k} c_i\ e^{-[(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)]^2/\sigma^2}\cdot \\
&\quad \cdot\tfrac{\partial}{\partial t}\{-[(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)]^2/\sigma^2\}.
\end{aligned} \tag{4}$$

Denote

$$\begin{aligned}
D &\triangleq \frac{\partial}{\partial t}[-(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)]^2/\sigma^2. \\
D &= -\frac{2}{\sigma^2}(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)\cdot \\
&\quad \cdot\tfrac{\partial}{\partial t}[(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i).
\end{aligned}$$

Since $\Delta\mathbf{x}$ is a unit vector, and by the commutativity of the inner product, we consequently have,

$$\begin{aligned}
&(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i) = \\
&\quad (\mathbf{x}_1-\mathbf{x}_i)^T(\mathbf{x}_1-\mathbf{x}_i) + 2t(\Delta\mathbf{x})^T(\mathbf{x}_1-\mathbf{x}_i),
\end{aligned}$$

and,

$$\frac{\partial}{\partial t}[(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)] = 2(\Delta\mathbf{x})^T(\mathbf{x}_1-\mathbf{x}_i) + 2t.$$

Thus,

$$D = -\frac{2}{\sigma^2}[\|\mathbf{x}_1-\mathbf{x}_i\| + 2t(\Delta\mathbf{x})^T(\mathbf{x}_1-\mathbf{x}_i)][2(\Delta\mathbf{x})^T(\mathbf{x}_1-\mathbf{x}_i) + 2t. \tag{5}$$

Consider the following two possible cases:

(A) $\forall i\ \ (\Delta\mathbf{x})^T(\mathbf{x}_1-\mathbf{x}_i) \geq 0$,

(B) $\exists i\ \ (\Delta\mathbf{x})^T(\mathbf{x}_1-\mathbf{x}_i) < 0$.

Note that case (B) means that the direction of change, determined by the vector $\Delta\mathbf{x}$ is along the view space spanned by the $\mathbf{x}_i,\ i = 1,\ldots k$, whereas in case (A), the direction of the displacement is orthogonal or away from the view space (see, again, Figure 11). Denote, $d_i \triangleq \|\mathbf{x}_1-\mathbf{x}_i\|$, $\Delta_i \triangleq (\Delta\mathbf{x})^T(\mathbf{x}_1-\mathbf{x}_i)$, and note that $d_i \geq 0$. With the new notation, equation (5) becomes,

$$\begin{aligned}
D &= -\frac{2}{\sigma^2}(d_i + 2t\Delta_i)(2\Delta_i + 2t) \\
&= -\frac{4}{\sigma^2}(d_i\Delta_i + d_i t + 2t{\Delta_i}^2 + 2t^2\Delta_i),
\end{aligned}$$

and when t goes to zero, this yields,

$$D \underset{t\to 0}{\longrightarrow} -\frac{4}{\sigma^2}d_i\Delta_i.$$

Consequently, in the limit for $t \to 0$, from equation (5) we have,

$$\frac{\partial RBF(\mathbf{x}+t\Delta\mathbf{x})}{\partial t}\,\Big|_{\substack{\mathbf{x}=\mathbf{x}_1\\ t>0,\ t\to 0}} \underset{t\to 0}{\longrightarrow} \sum_{i=1}^{k} c_i\ e^{-\frac{{d_i}^2}{{\Delta_i}^2}} \cdot \left(-\frac{4}{\sigma^2}d_i\Delta_i\right). \tag{6}$$

Denote this limit by L, $L = -\frac{4}{\sigma^2}\sum_{i=1}^{k} c_i d_i \Delta_i \; e^{-\frac{d_i{}^2}{\Delta_i{}^2}}$.
For case (A), $\Delta_i \geq 0, \;\; \forall i$; Therefore, if all $c_i > 0$, we would have $L_A \leq 0, \; L_A < L_B$, for $L_A, \; L_B$ the values of the limit L, for cases (A) and (B), respectively. This means that an infinitesimal displacement along the view space results in a smaller change of the corresponding RBF activity than the RBF change resulted from a displacement that is orthogonal to, or away from the view space. This establishes the desired property of an RBF-based classifier — an approximate constant behavior for different views of the target shape, with the response falling off for views of different shapes — for the infinitesimal view change case.

Claim A.1 $c_i > 0, \;\; \forall i = 1, \ldots, k$.

Proof:
From equation (2) we have $c_i = \sum_j (\mathbf{G}^+)_{ij} y_j$, the sum of elements in the i^{th} row of the matrix $\mathbf{G}^+$, where y_j are the targets, $y_j = 1, \; j = 1, \ldots k$, and $\mathbf{G}^+$ is the (pseudo) inverse of $\mathbf{G}$ whose elements are $\mathbf{G}_{ij} = e^{-d_{ij}^2/\sigma^2}$, for $d_{ij} \triangleq \|\mathbf{x}_i - \mathbf{x}_j\|$. Note that $\mathbf{G} = I + A$, where I is a unit matrix[5], and A is a matrix whose elements are $\ll 1$, under a proper bound on σ (see below). Thus, by Taylor expansion for the matrix $\mathbf{G}$, we have,

$$\mathbf{G}^+ = \frac{1}{I + A} \approx I - A + O(A^2).$$

To complete the proof, let $\sigma < (\ln k)^{-1/2} \min_{\substack{i,j \\ i<j}} d_{ij}$, for k - the number of training vectors. Thus, for all i and j, $d_{ij} > \sigma (\ln k)^{1/2}$, $d_{ij}^2 > \sigma^2 \ln k$, and $-\frac{d_{ij}^2}{\sigma^2} < -\ln k = \ln \frac{1}{k}$. Taking the exponent of both terms, we obtain

$$e^{-\frac{d_{ij}^2}{\sigma^2}} < e^{\ln \frac{1}{k}} = \frac{1}{k}.$$

As a result, the sum of elements in any row of $\mathbf{G}^+$ consists of 1 (the element on the diagonal, contributed by the unit matrix) minus $k-1$ elements, each smaller than $\frac{1}{k}$. Thus, we finally have,

$$\forall i = 1, \ldots, k,$$
$$c_i \;\; = 1 - \sum_{j=1}^{k-1} e^{-\frac{d_{ij}^2}{\sigma^2}} y_j > 1 - \sum_{j=1}^{k-1} \tfrac{1}{k} = 1 - \tfrac{k-1}{k} > 0.$$

A.2 The finite displacement case

We next extend the above proof to a finite view displacement. As before, we consider a change in object appearance due to (a) the extrinsic effect of pose, i.e. a change along view space direction (object rotation), and (b) an intrinsic shape change, that is, a change orthogonal to, or away from the view space (shape deformation).
First, note that the two factors determining the two-dimensional appearance of an object, the shape and pose, are orthogonal. To demonstrate this, we have simulated shape and pose variation for three-dimensional objects consisting of a collection of points in 3D. For such a point-cloud object, shape deformation is simulated by a random displacement of the cloud's points, whereas a change of pose simply means an arbitrary rotation of all points. The two-dimensional appearance of the deformed, or rotated object is obtained by an orthographic projection, and the displacement from the two-dimensional appearance of the original cloud is measured. The inner product between the two vectors, representing the changes in appearance caused by rotation and deformation, is calculated to find the cosine of the angle between the shape and pose displacements. Figure 12 shows the above calculation for different combinations of shape and pose variations, averaged over many independent runs. Indeed, for a significant range of variation, orthogonality is observed between the shape and pose factors that determine the appearance of an object.
Now, let $\mathbf{x}_1$ be, as before, an arbitrary training view of the object, and let Δv, Δp, be finite displacements along, and in perpendicular to view space, respectively.
Note that because Gaussians are factorizable, and because the view-space and the shape-space projections of an object appearance are orthogonal to each other, we have

$$G(\|\mathbf{x} - \mathbf{t}\|) = e^{\frac{-\|\mathbf{x}-\mathbf{t}\|^2}{\sigma^2}} = e^{\frac{-\|\mathbf{x}^p - \mathbf{t}^p\|^2}{\sigma^2}} e^{\frac{-\|\mathbf{x}^v - \mathbf{t}^v\|^2}{\sigma^2}}. \tag{7}$$

Consider now a displacement within an object view space. This change in the object's (two-dimensional) appearance results from a (three-dimensional) rotation of the object away from some reference view. The upper bound on this kind of change is therefore finite. To see that, recall that both $\{\mathbf{x}_i\}_{i=1}^N$ and $\mathbf{x}$ are different two-dimensional views of the same object, resulting from projection of the corresponding three-dimensional "views," $\mathcal{X}_i, \; i = 1 \ldots, k$, and $\mathcal{X}$,

[5] $\forall i, \; d_{ii} = 0$, thus, $e^{-d_{ii}^2/\sigma^2} = 1$ are the diagonal elements.

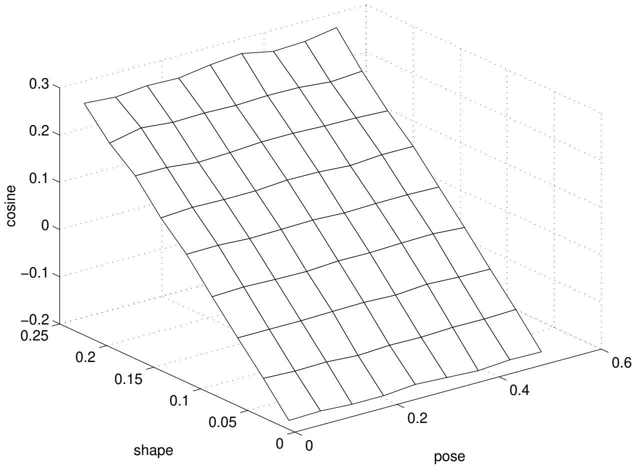


Figure 12: Orthogonality of shape and pose. The displacement in the two-dimensional appearance of a three-dimensional 10-point cloud object due to variations in pose and shape is measured, assuming orthographic projection. The plot shows the average value of the cosine of the angle between the shape and pose displacements, calculated for 20,000 randomly chosen values of pose variation (an arbitrary rotation of the cloud's points), and shape deformation (a random displacement of the cloud's points). Data were gathered into a small number of bins, sorted by the angle of rotation (shown in radians along the *pose* axis), and by the amount of shape deformation, measured as the fraction of the random displacement with respect to the total cloud distribution (*shape* axis).

respectively. That is, $\mathbf{x} = \mathcal{P}\mathcal{X}$, $\mathbf{x}_i = \mathcal{P}\mathcal{X}_i$, where $\mathcal{P}$ is a $3D \longrightarrow 2D$ projection. Any three-dimensional view can be described by an object rotation $R_{\mathbf{n}}(\omega)$ away from some orientation, say $\mathcal{X}_c$ in the three-dimensional space.
Thus,

$$\begin{aligned} &\|\mathbf{x} - \mathbf{x}_i\| = \\ &\quad \|\mathcal{P}\mathcal{X} - \mathcal{P}\mathcal{X}_i\| = \|\mathcal{P}R_{\mathbf{n}_1}(\omega_1)\mathcal{X}_c - \mathcal{P}R_{\mathbf{n}_i}(\omega_i)\mathcal{X}_c\|. \end{aligned}$$

Under orthographic projection, the difference between projected vectors is the projection of their difference,and the norm which can only be reduced by projection, is preserved by the rotation mapping (Kanatani, 1990). Thus,

$$\begin{aligned} &\|\mathcal{P}[R_{\mathbf{n}_1}(\omega_1)\mathcal{X}_c - R_{\mathbf{n}_i}(\omega_i)\mathcal{X}_c]\| \leq \\ &\quad \|R_{\mathbf{n}_1}(\omega_1)\mathcal{X}_c - R_{\mathbf{n}_i}(\omega_i)\mathcal{X}_c\| \leq \\ &\quad \|R_{\mathbf{n}_1}(\omega_1)\mathcal{X}_c\| + \|R_{\mathbf{n}_i}(\omega_i)\mathcal{X}_c\| = \\ &\quad \|\mathcal{X}_c\| + \|\mathcal{X}_c\| = 2\,\|\mathcal{X}_c\|. \end{aligned}$$

Thus, an upper bound on the extent of the view space displacement is easily established. We denote this bound by D. Let $\mathbf{x} = \mathbf{x}_1 + \Delta v$. From the above, $\|\Delta v\| \leq D$. By triangle inequality,

$$\begin{aligned} \|\mathbf{x} - \mathbf{x}_i\| = \|(\mathbf{x}_1 + \Delta v) - \mathbf{x}_i\| \leq \\ \|(\mathbf{x}_1 + \Delta v) - \mathbf{x}_1\| + \|\mathbf{x}_1 - \mathbf{x}_i\| = \|\Delta v\| + \|\mathbf{x}_1 - \mathbf{x}_i\|. \end{aligned}$$

Hence,

$$-\|\mathbf{x} - \mathbf{x}_i\|^2 \geq -\left[\|\Delta v\|^2 + 2\|\Delta v\| \cdot \|\mathbf{x}_1 - \mathbf{x}_i\| + \|\mathbf{x}_1 - \mathbf{x}_i\|^2\right].$$

As a consequence, because all c_i are positive (Claim A.1),

$$RBF(\mathbf{x}) = \sum_{i=1}^{k} c_i\, e^{-\|\mathbf{x}-\mathbf{x}_i\|^2/\sigma^2} \geq$$

$$\sum_{i=1}^{k} c_i\, e^{-\|\Delta v\|^2/\sigma^2} \cdot e^{-2\|\Delta v\| \cdot \|\mathbf{x}_1 - \mathbf{x}_i\|/\sigma^2} \cdot e^{-\|\mathbf{x}_1-\mathbf{x}_i\|^2/\sigma^2}.$$

Now, let $\sigma < 2 \min\limits_{\substack{i,j \\ i<j}} d_{ij}$, for $d_{ij} \triangleq \|\mathbf{x}_i - \mathbf{x}_j\|$.
Thus, $\|\mathbf{x}_1 - \mathbf{x}_i\| \geq \frac{\sigma}{2}$.
Because $\|\Delta v\| \leq D$, and $-2\,\|\Delta v\| \geq -2D$, we have,

$$\frac{-2\|\Delta v\|\|\mathbf{x}_1 - \mathbf{x}_i\|}{\sigma^2} \geq -\frac{D}{\sigma}.$$

Finally,

$$RBF(\mathbf{x}) \geq \sum_{i=1}^{k} c_i\; e^{-\|\mathbf{x}_1 - \mathbf{x}_i\|^2/\sigma^2} \cdot e^{-\frac{D^2}{\sigma^2}} \cdot e^{-\frac{D}{\sigma}},$$

or,

$$RBF(\mathbf{x}) \geq e^{-\frac{D}{\sigma}(1 + \frac{D}{\sigma})} \cdot RBF(\mathbf{x}_1),$$

for

$$D \ll \sigma, \;\; F \triangleq \frac{D}{\sigma} \ll 1,$$

and,

$$e^{-F(1+F)} \gg 0.$$

Now, for a finite displacement in perpendicular to the view space, $\mathbf{x} = \mathbf{x}_1 + \Delta p$, we have by orthogonality (equation (7)),

$$RBF(\mathbf{x}) = \sum_{i=1}^{k} c_i\; e^{-\|(\mathbf{x}_1 + \Delta p) - \mathbf{x}_i\|^2/\sigma^2} =$$

$$\sum_{i=1}^{k} c_i\; e^{-\|\mathbf{x}_1 - \mathbf{x}_i\|^2/\sigma^2} \cdot e^{-\|\Delta p\|^2/\sigma^2} = RBF(\mathbf{x}_1) \cdot e^{-\|\Delta p\|^2/\sigma^2}.$$

For an arbitrary amount of shape-space displacement, say, $\Delta p \gg 0$, $e^{-\|\Delta p\|^2/\sigma^2} \ll 1$ can become arbitrarily small, since $-\Delta p^2 \ll 0 \Longrightarrow e^{-\|\Delta p\|^2/\sigma^2} \ll 1$.
Hence we finally have, for a shape displacement,

$$RBF(\mathbf{x}) \leq e^{-\|\Delta p\|^2/\sigma^2} RBF(\mathbf{x}_1) \ll RBF(\mathbf{x}_1).$$

From the above arguments, we may conclude that (1) any displacement *within* the view space of the target object results in an RBF activity that cannot be less than some positive, not too small, fraction of its activity on the training examples, whereas (2) for a displacement *in perpendicular* to the view space, the corresponding RBF activity is always below the activity obtained in training, with the activity decreasing for increasing shape differences.

B Training individual shape-specific modules

To train an RBF module one needs to place the basis functions optimally as to cover the input space (i.e., determine the basis-function centers), calculate the output weights associated with each center, and tune the basis-function width.

B.1 Finding the optimal placement for each basis function

Whereas the computation of the weight assigned to each basis function is a linear optimization problem, finding the optimal placement for each basis in the input space is much more difficult (Poggio and Girosi, 1990). Here, we consider a simplified version of this problem, which assumes that a small optimal subset of examples to be used in training is chosen out of a larger set of available data, consisting of views of the shape on which the module is trained. Views are given by their measurement-space representations (here, we used a small collection of filters with radially elongated Gaussian receptive fields, randomly positioned over the image (Weiss and Edelman, 1995); see Figure 4). This approach leads naturally to the question of the definition of optimality. Defining an optimal subset of views as the subset that minimizes the nearest-neighbor classification error amounts to performing vector quantization (VQ; see appendix C) in the input space (Moody and Darken, 1989; Poggio and Girosi, 1989).
By definition, quantizing an input space results in a set of vectors that are the best representation of the entire space. A quantization is said to be optimal if it minimizes an expected distortion. Simple measures of the latter, such as squared Euclidean distance, while widely used in vector quantization applications (Gersho and Gray, 1992), do not correlate well with the subjective notion of distance appropriate for the task of quantizing an object view space. Specifically, Euclidean distances in a pixel space do not reflect object identities if the illumination conditions are allowed to vary (Adini et al., 1997). Likewise, in a Euclidean receptive-field (RF) space, images of similar objects tend to cluster together by view, not by object shape, if objects may rotate (Duvdevani-Bar and Edelman, 1995;

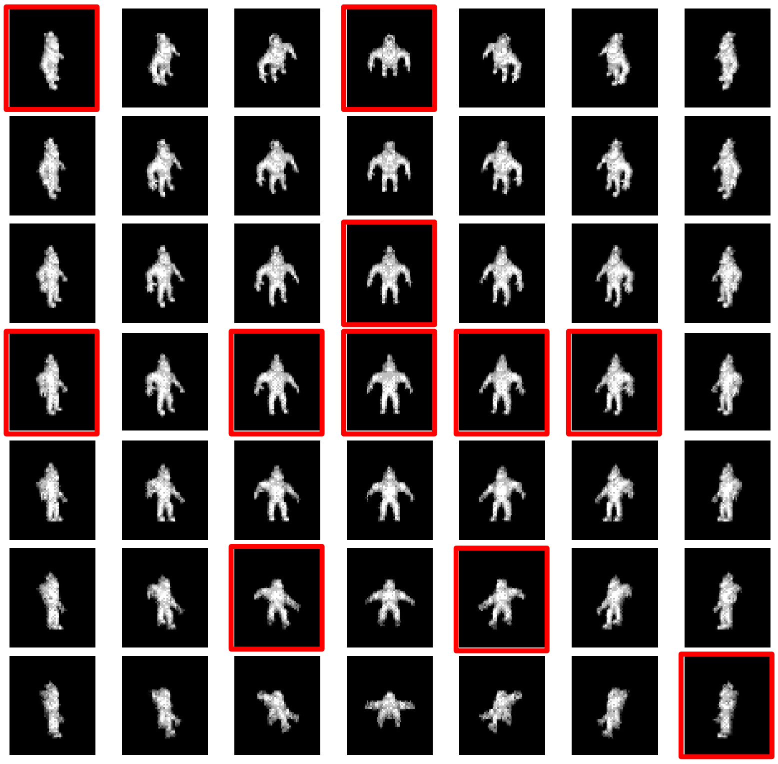


Figure 13: A set of 49 views of one of the figure-like test objects (`A1`), taken at grid points along an imaginary viewing sphere centered around the object. Views differ in the azimuth and the elevation of the camera, both ranging between $-60°$ and $60°$ at $20°$ increments. We used the Canonical Vector Quantization (CVQ) procedure to select the most representative views for the purpose of training the object representation system (section B.1; the selected views of `A1` are marked by frames.

Lando and Edelman, 1995). This implies that Euclidean distance between RF representation of object views cannot overcome the variability in object appearance caused by changes in viewing conditions, and that a different measure of quantization distortion is needed.

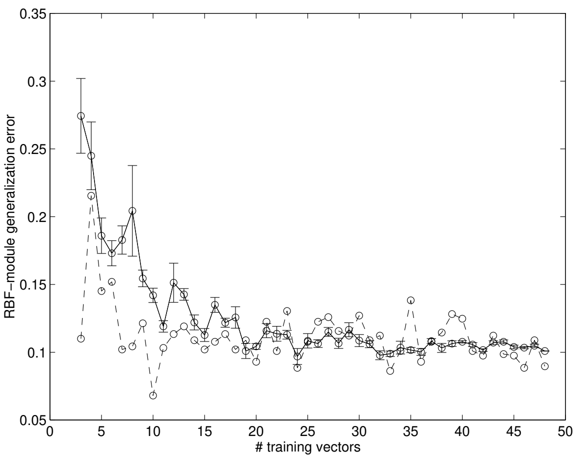
The measure we incorporated in the present model is canonical distortion, proposed by Baxter (1996). The notion of canonical distortion is based on the observation that in any given classification task, there exists a natural environment of functions, or classifiers, that allow for a faithful representation of distance in the input space. The property shared by all such classifiers is that their output varies little across instances of the same entity (class); ideally, the output of a particular classifier is close to one if the input is an instance of its target class, and is close to zero otherwise. Thus, in the space of classifier *outputs* instances of the same class are closer together, and instances of different classes farther apart, than in the input space. According to Baxter, the distortion measurement induced by the classifier space is the desired canonical distortion measure.[6]

Following Baxter's ideas, we sample the view space of an object at a fixed grid wrapped around the viewing sphere centered at the object (see Figure 13), then *canonically* quantize the resulting set of object views. The representative views, which are subsequently used to train the object-specific modules, are chosen in accordance with the following three criteria. First, a classifier (i.e., module) output should be approximately constant for different views of its selected object. Second, views of the same object should be tightly clustered in the classifier output space. Third, clusters corresponding to views of different objects should be separated as widely as possible.

We have combined these three criteria in a modified version of the Generalized Lloyd algorithm (GLA) for vector quantization (Linde et al., 1980), known also as the k-means method (MacQueen, 1967). In contrast to the conventional GLA, which carries out quantization in the *input* vector space, our algorithm concentrates on the classifier *output* space. Training an RBF network on the centers of clusters resulting from the optimal partition of the classifier output space addresses the first of the three requirements — an approximately constant output across views of an object. The other two requirements are addressed by a simultaneous minimization of the ratio of between-objects to within-object view scatter (a cluster compactness criterion; see Duda and Hart, 1973).

[6]Formally, for an environment of functions $f \in \mathcal{F}$, mapping a probability space (X, P, σ_X) into a space (Y, σ), with $\sigma : Y \times Y \to R$, a natural distortion measure on X, induced by the environment is $\rho(x, y) = \int_{\mathcal{F}} \sigma(f(x), f(y)) dQ(f)$, for $x, y \in X$, and Q an environmental probability measure on $\mathcal{F}$.

Increasing the number of examples on which a classifier is trained always improves both the RBF-module classifier performance and the view-space compactness criterion (see Figure 14). Our version of Baxter's Canonical Vector Quantization (CVQ) relies on this observation by taking the so-called "greedy" algorithmic approach. The algorithm is initialized with an empty set of views and adds new views iteratively. At each iteration, the new view is chosen so as to minimize the compactness criterion, and the entire process follows the gradient of improvement in classifier performance (see appendix C.1, for details).



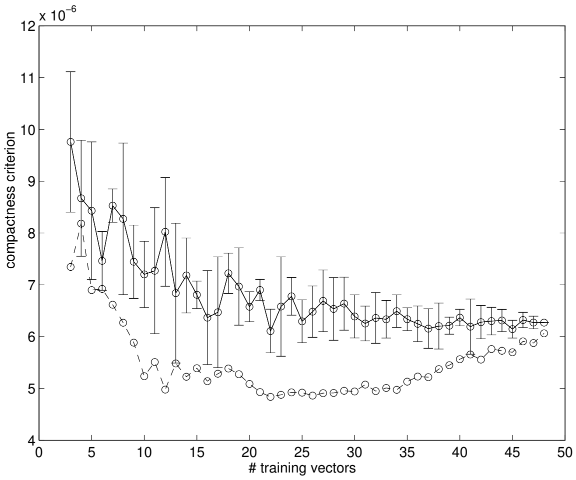


Figure 14: The effect of training-set size on the performance of an RBF module trained under the compactness criterion. *Left:* the recognition error obtained for the Nissan module, trained as a part of a network consisting of ten object modules (see Figure 5 below). For each object, training involved a set of $N = 49$ views, taken as described in Figure 13. The abscissa is the number t of the training vectors (examples). For $t < 15$ or so, the performance of the module trained on the CVQ-derived *code vectors* (dashed line) is better than the error obtained with the same number of randomly chosen training vectors (solid line). When t is large, the resulting error is low in any case. *Right:* The compactness criterion (the ordinate), defined as the ratio of between-cluster to within-cluster scatter (Duda and Hart, 1973), plotted against the size of the training set. Note that the values of the compactness criterion obtained for the CVQ code vectors (dashed line) are significantly better (lower) than the values obtained for a module trained on the same number of randomly chosen vectors (solid line). In both plots, the error bars represent the standard error of the mean, calculated over 25 independent random choices of the training vectors.

B.2 Tuning the basis-function width

A complete specification of an RBF module consists of the choice of basis function centers, the output weights associated with each center, and the spread constant, or the width, of the basis functions. The width parameter has a direct influence on the performance of an RBF classifier (i.e., its ability to accept instances of the class on which it is trained and to reject other input). Optimally, the width parameter should be set to a value that yields equal miss and false-alarm error rates (see Figure 15). Following the rule of thumb according to which the width parameter should be much larger than the minimum distance and much smaller than the maximum distance among the basis centers, we employ a straightforward binary search to optimize its value.

C Vector quantization

Vector quantization (VQ) is a technique that has been originally developed for signal coding in communications and signal processing. It is used in a variety of tasks, including speech and image compression, speech recognition and signal processing (Gersho and Gray, 1992).
A vector quantizer Q is a mapping from a d-dimensional Euclidean space, $\mathcal{S}$, into a finite set $\mathcal{C}$ of *code vectors*, $Q : \mathcal{S} \to \mathcal{C}$, $\mathcal{C} = (p_1, p_2, \ldots, p_n), p_i \in \mathcal{S}, i = 1, 2, \ldots, n$. Associated with every n-point vector quantizer is a partition of $\mathcal{S}$ into n regions, $R_i = \{x \in \mathcal{S} : Q(x) = p_i\}$.
Vector quantizer performance is measured by distortion $d(\mathbf{x}, \hat{\mathbf{x}})$ — a cost associated with representing an input vector $\mathbf{x}$ by a quantized vector $\hat{\mathbf{x}}$. The goal in designing an optimal vector quantization set is to minimize the expected distortion. The most convenient and widely used measure of distortion is the squared Euclidean distance.

C.1 The generalized Lloyd (K-means) algorithm

The generalized Lloyd algorithm (GLA) for vector quantizer design (Linde et al., 1980) is known also as the k-means method (MacQueen, 1967). According to the algorithm, an optimal vector quantizer is designed via iterative

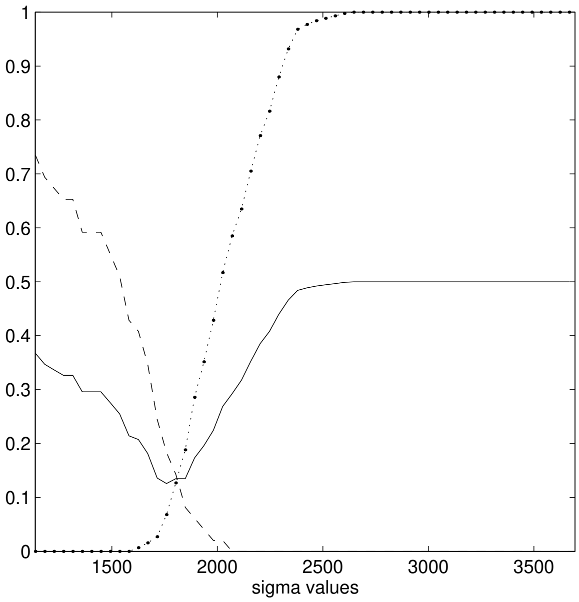


Figure 15: The effect of the basis function width (σ) on the performance of an RBF module. *Left:* RBF-module miss rate (dashed line), false-alarm rate (dotted line) and their mean (solid line), plotted against σ. The values of σ shown on the abscissa range from half the minimal distance up to the maximal distance among RBF-module "centers" (training views) in the input space.

codebook modifications to satisfy two conditions: nearest neighbor (NN) and centroid condition (CC). The former is equivalent to constructing the Voronoi cell of each code vector, whereas the application of the latter is aimed to adjust each code vector to be the center of gravity of its domination region. The means of the (k) initial clusters are found, and each input point is examined to see if it is closer to the mean of another cluster than it is to the mean of its current cluster. In that case, the point is transferred and the cluster means (centers) are recalculated. This procedure is repeated until the chosen measure of distortion is sufficiently small.

C.2 The Lloyd algorithm modified to perform canonical quantization

We next present our modification of the GLA for the canonical vector quantization (CVQ) design.

1. Initialization: Set $N = 2$, an initial codebook size. Set $E_N = \infty$. Set $\mathcal{C}^N$ to be an initial codebook of size N. The codebook is randomly chosen from the input set.
2. Find an input vector for which the compactness is optimal, and add it to $\mathcal{C}^N$ to create a codebook $\mathcal{C}^{N+1}$ of size $N + 1$.
   (a) Set iteration $m = 1$, $D_m = \infty$.
   (b) Given the codebook $\mathcal{C}^N_m$, perform the modified Lloyd Iteration on the classifier output space to generate the improved codebook $\mathcal{C}^N_{m+1}$.
   (c) Compute the sum-of-squared-error D_m. If $\frac{D_m - D_{m+1}}{D_m} < \epsilon$ for a suitable threshold ϵ, halt. The improved codebook $\mathcal{C}^N_{m+1}$ is the set of *input* vectors, whose classifier *outputs* are the closest to the codevectors constituting the improved output codebook (see below).
   Otherwise, set $m \leftarrow m + 1$, go to Step (b).
3. Calculate the classifier generalization error E_N. If the criterion $\frac{E_N - E_{N+1}}{E_N} \leq \epsilon$ is satisfied, finish. Otherwise, set $N \leftarrow N + 1$, go to Step (2).

The modified Lloyd Iteration:

1. Compute classifier activity over the input set, denote this set by $\mathcal{O}$. Denote the set of classifier outputs on the codebook $\mathcal{C}^N_m$, the *output codebook.*
2. Partition the set $\mathcal{O}$ into clusters using the *Nearest Neighbor Condition*, for the output-codebook vectors being the cluster centers.
3. Using the *Centroid Condition*, compute the centroids for the clusters just found, to obtain a new output codebook.

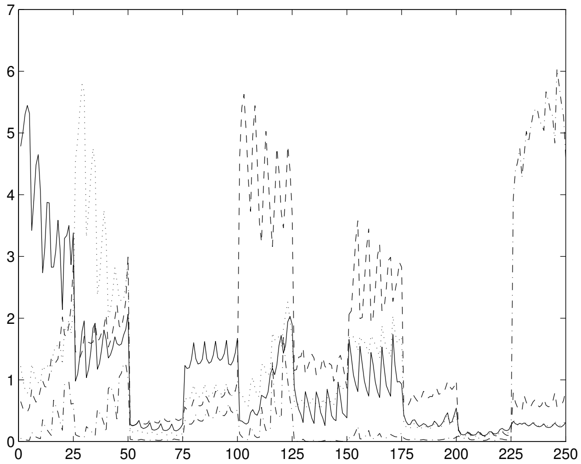


Figure 16: The activity of several RBF modules obtained for 100 test views (25 views for each of four objects). The views, which vary along the abscissa, are grouped, so that the first 25 views belong to the first object (`cow`, solid line), with the subsequent views, in groups of 25, belonging, respectively, to `cat` (dotted line), `tuna` (dashed line), and `TRex` (dash-dotted line). Note that each classifier responds strongly to views of its target object, and significantly less to views of other objects.

D Additional tables

	cow1	cat	Al	gene	tuna	Lrov	Niss	F16	fly	TRex
cow1	**4.04**	1.86	0.42	1.62	0.91	1.22	1.79	1.21	0.71	0.53
cat2	1.69	**3.55**	0.26	1.02	1.10	1.20	2.10	1.04	0.61	0.53
Al	0.08	0.06	**1.63**	0.46	0.03	0.12	0.06	0.09	0.19	0.06
gene	0.61	0.43	0.44	**5.24**	0.14	0.11	0.26	0.48	0.55	0.25
tuna	1.57	2.00	0.40	1.11	**4.22**	1.41	3.05	1.77	0.72	1.02
Lrov	0.57	0.56	0.17	0.20	0.23	**3.36**	1.38	0.36	0.16	0.11
Niss	0.67	0.86	0.06	0.34	0.82	0.97	**3.24**	0.88	0.21	0.25
F16	0.50	0.44	0.11	0.65	0.58	0.27	0.94	**2.14**	0.24	0.25
fly	1.03	1.08	0.88	2.30	0.60	0.70	0.95	0.84	**3.71**	0.99
TRex	0.28	0.34	0.09	0.60	0.32	0.14	0.44	0.36	0.29	**3.67**

Table 4: RBF module activities (averaged over all 169 test views) evoked by the trained objects. Each row shows the average activation pattern induced by views of one of the objects over the ten reference-object RBF modules; boldface indicates the largest entry (see section 4.1).

	cow1	cat2	Al	Gene	tuna	Lrov	Niss	F16	fly	TRex
frog	0.38	0.28	0.29	0.18	0.35	0.20	0.11	0.09	**0.99**	0.16
turtle	**0.53**	0.32	0.38	**0.64**	0.39	0.13	0.09	0.13	**0.93**	0.17
shoe	0.51	**0.63**	0.06	0.12	**1.09**	0.46	0.54	0.33	**0.59**	0.16
pump	**1.33**	**1.44**	0.01	0.17	**2.37**	0.32	1.02	0.40	0.83	0.19
Beetho	0.09	0.05	0.10	0.02	0.07	0.05	0.01	0.01	**0.38**	0.01
girl	**2.66**	**1.78**	0.13	**3.27**	**2.55**	0.20	0.73	1.07	**2.03**	0.86
lamp	0.72	0.48	0.71	0.70	0.41	0.36	0.09	0.09	**1.53**	0.09
manate	**1.49**	0.98	0.09	0.36	**2.47**	0.35	**1.45**	0.68	0.84	0.24
dolphi	**1.14**	0.98	0.04	0.34	**2.20**	0.23	0.68	0.51	0.72	0.13
Fiat	1.51	1.77	0.01	0.12	**3.76**	0.46	**2.27**	0.87	0.79	0.27
Toyota	**2.16**	**2.13**	0.10	0.25	**2.50**	**2.00**	**2.29**	0.69	0.83	0.30
tank	**1.85**	**1.91**	0.09	0.51	**2.50**	1.04	**2.36**	**1.46**	1.08	0.56
Stego	**2.04**	**2.13**	0.06	0.67	**3.61**	0.67	**2.45**	1.46	1.58	0.98
camel	**2.20**	**1.34**	0.04	0.77	**1.75**	0.30	0.65	0.54	1.02	0.23
giraff	**1.87**	**1.93**	0.03	0.54	**3.24**	0.19	1.04	1.21	**1.63**	**1.72**
Gchair	**1.75**	**1.69**	0.00	0.09	**3.04**	0.29	1.40	0.76	0.86	0.19
chair	**2.64**	**2.65**	0.02	0.44	**4.05**	0.82	2.39	1.06	1.78	0.51
shell	**1.89**	**1.09**	0.25	**1.56**	**0.95**	0.44	0.40	0.49	**1.66**	0.35
bunny	**1.07**	**1.24**	0.23	0.22	**1.10**	**1.47**	0.53	0.28	**0.95**	0.30
lion	**0.55**	**0.59**	0.09	0.13	**0.54**	**0.61**	0.20	0.09	**0.60**	0.13

Table 5: RBF activities (averaged over all 169 test views) for the 20 test objects shown in Figure 9. In each row (corresponding to a different test object), entries within 50% of the maximum for that row are marked by boldface. These entries constitute a low-dimensional representation of the test object whose label appears at the head of the row, in terms of similarities to some of the ten reference objects. For instance, the `manatee` (an aquatic mammal known as the sea cow) turns out to be like (in decreasing order of similarity), a `tuna`, a `cow`, and, interestingly, but perhaps not surprisingly, a `Nissan` wagon.

	obj	cow1	cat2	Al	gene	tuna	Lrov	Niss	F16	fly	TRex
QUAD	cow2	0.69	0.30				0.01				
	ox	0.93	0.04	0.02	0.02						
	calf	0.86	0.06			0.06		0.01	0.02		
	deer	0.34	0.62			0.03		0.01			
	Babe	0.88	0.05				0.04			0.03	
	PigMa	0.83	0.12					0.02		0.04	
	dog	0.33	0.64			0.01		0.01	0.01		
	goat	0.20	0.69	0.04	0.06					0.02	
	buff	0.72	0.17		0.03	0.01	0.03			0.05	
	rhino	0.69	0.15			0.01	0.02	0.11	0.01		
FIGS	pengu	0.30	0.11		0.28			0.01	0.01	0.29	
	ape	0.11	0.11	0.31						0.47	
	bear	0.08	0.07		0.75			0.01		0.10	
	cands		0.16	0.74						0.10	
	king			0.67	0.09					0.24	
	pawn			0.73						0.27	
	venus			0.86	0.01					0.13	
	lamp	0.04		0.64			0.04			0.28	
	lamp2	0.03		0.70						0.27	
	lamp3			0.70	0.14					0.17	
FISH	whale	0.08	0.11			0.80			0.01		
	whalK	0.04	0.04			0.91				0.01	
	shark	0.03	0.07			0.89					0.01
	Marln		0.01			0.98		0.01			
	whalH	0.10	0.20			0.70					
AIR	F15	0.12	0.08			0.02		0.02	0.72		0.03
	F18	0.09	0.07			0.06		0.01	0.78		
	Mig27	0.05	0.37	0.14		0.12			0.31		
	shutl	0.24	0.31			0.30			0.13		0.02
	Ta4	0.11	0.17			0.10		0.02	0.55		0.05
CARS	Isuzu	0.07	0.07				0.04	0.83			
	Mazda	0.04	0.07				0.01	0.88			
	Mrcds	0.04	0.04					0.92			
	Mitsb	0.04	0.07				0.01	0.89			
	NissQ	0.07	0.08				0.01	0.83		0.01	
	Subru	0.04	0.04					0.92			
	SuzuS	0.13	0.17			0.08	0.30	0.33			
	ToyoC	0.09	0.07				0.05	0.79			
	Beetl	0.03	0.09					0.87		0.01	
	truck	0.07	0.05					0.89			
DINO	Paras	0.01	0.05			0.01					0.93
	Veloc		0.03			0.24			0.02		0.71
	Allos		0.21			0.36		0.04	0.02		0.36

Table 6: Categorization results for the 43 test objects shown in Figure 6, for the k-NN method of section 4.2.4, with $k = 3$. Each row corresponds to one of the test objects; the proportion of the 169 test views of that object attributed to each of the categories present in the training set appears in the appropriate column. Note that the misclassification rate depends on the definition of category labels. Here, mean misclassification rate, over all 169 views of all objects, was 22% for the first set of category labels (i.e., the seven categories illustrated in Figure 5), 16% for the second set of labels (according to which the fly and the FIGURES have the same label), and 14% for the third set of labels (where in addition the tuna and the F16 have the same category label).

♯ Test Objs	♯ Reference Objs				
	1	5	10	15	20
2	0	0	0	0	0
5	0.077	0.011	0.006	0.008	0.006
10	0.140	0.024	0.009	0.008	0.007
25	0.183	0.026	0.009	0.005	0.005
50	0.055	0.022	0.012	0.008	0.007

Table 7: Error rate obtained for the discrimination task vs. the number of test and reference objects (these data are also plotted in Figure 10). The error rate in entry (Np, Nt) is the mean error rate obtained for the discrimination task using the activities of Np reference objects, and tested on 25 views of each of the Nt test objects, employing the 3-NN procedure of section 4.2.2. The mean is taken over 10 independent choices of Np objects out of 20 available reference objects, and 10 random selections of Nt objects out of a set consisting of 50 test objects (total of $(5 \cdot 10)(5 \cdot 10) = 2500$ independent trials).