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

Sharon Duvdevani-Bar
Department of Applied Mathematics,
The Weizmann Institute of Science,
Rehovot, 76100, Israel

Shimon Edelman*
School of Cognitive and Computing Sciences,
University of Sussex,
Falmer, Brighton BN1 9QH, UK
shimone@cogs.susx.ac.uk

January 1998
revised November 1998, April 1999

Abstract

One of the difficulties of object recognition stems from the need to overcome the variability in object appearance caused by pose and other factors, such as illumination. 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. Although categorization cannot rely on interpolation between stored examples, we show that knowledge of several representative members, or prototypes, of each of the categories of interest can provide the necessary computational substrate for the categorization of new instances. We describe a system that represents input shapes by their similarities to several prototypical objects, and show that it can recognize new views of the familiar objects, discriminate among views of previously unseen shapes, and attribute the latter to familiar categories.

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

*To whom correspondence should be addressed.

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 possible to extend such a memory-based strategy to deal with *categorization*, a task that requires the system to make sense of novel shapes. It turns out that familiarity with a relatively small selection of objects can be used as a foundation for processing other objects, never seen before. Specifically, objects (including novel ones) can be effectively described in terms of their similarities to a relatively small number of reference shapes, for each of which the system maintains a view-based recognition module.

Dealing with novel shapes (over and above novel views of familiar shapes) is a challenging problem, which we chose to approach separately from other difficulties arising in object recognition. We assume, therefore, that the objects are pre-segmented from the background, and that their size, location in the image, and the illumination parameters are held constant. With this assumption in mind, let us compare the computational requirements of the two classes of shape-related tasks mentioned earlier: recognition and categorization.

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. One way to do that is by gaining information about these effects from multiple images of each object, corresponding to different values of factors such as viewpoint.

A simple model of recognition that adopts this approach 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). We refer to this manifold as the *view space* of the object. More formally, let a *3D-view* (Poggio and Vetter, 1992) of object A at orientation θ be defined by the coordinates of k fiducial points:

$$\mathbf{v}_A(\theta) = (x_1, y_1, z_1, \ldots, x_k, y_k, z_k)^T \tag{1}$$

In addition to the trivial observation that $\mathbf{v}_A \in R^{3k}$, one can prove that $\forall\theta, \mathbf{v}_A(\theta) \in \mathcal{V}_A \subset R^9$, where the view space $\mathcal{V}_A$ is a manifold whose dimensionality depends on that of θ and is equal to 2 if the object is allowed to rotate in depth but not undergo any other transformation (Ullman and Basri, 1991).[1]

Obviously, no visual recognition system has direct access to 3D-views of objects (that is, to their geometry). At the very least, two kinds of transformations interpose between object geometry and the recognition stage: imaging (which includes projection) and measurement. In machine vision, the outcome of the latter step is an array of pixel values, $P \in \mathcal{P}^{(q)}_{m\times n}$, where $\mathcal{P}^{(q)}_{m\times n}$ is the set of all $m \times n$ images quantized at q levels. In some systems (including the one described in this paper), the pixel array is piped through a bank of filters mimicking biological receptive fields, before any recognition is attempted. Either the raw pixel array or the filter outputs can be regarded as the *measurement space* on which subsequent processes operate.

Typically, the mapping M from the object geometry to the measurement space, $M : \mathcal{V}_A \rightarrow \mathcal{P}^{(q)}_{m\times n}$, is well-behaved, in the sense that it maps the smooth low-dimensional *distal* view space manifold into a smooth and equally low-dimensional *proximal* view space manifold, $\mathcal{V}^{(p)}_A \subset \mathcal{P}^{(q)}_{m\times n}$.[2] This property of the mapping M stems from the well-behavedness of its components: the relationship between the coordinates of the fiducial points and the local surface orientation, the effect of surface orientation on local image intensity, the registration of light in the photosensitive elements in the camera, *etc.* (Edelman and Duvdevani-Bar, 1997b).

The operational consequence of the smoothness of the proximal view space is that a new view of object A, which gets mapped into some member of that space $\mathbf{v}^{(p)}_A \in \mathcal{V}^{(p)}_A$, can be recognized by

[1] A more precise specification of the dimensionality and the shape of $\mathbf{v}_A$, which goes beyond the needs of the present discussion, can be found in (Ullman and Basri, 1991; Poggio and Vetter, 1992).

[2] The qualifier *distal* refers to objects "out there" in the world; the space $\mathcal{V}_A$ of which $\mathbf{v}_A$ (see eq. 1) is a member is the distal view space. The qualifier *proximal*, denoted by the superscript (p), refers to entities that reside in the measurement space, such as the manifold $\mathcal{V}^{(p)}_A$.

interpolation among its selected stored views, $\left\{\mathbf{v}_i^{(p)}\right\}$, $i = 1, \ldots, s$, where s is the number of stored views (the subscript A has been dropped for clarity). As observed by (Poggio and Edelman, 1990), a criterion that indicates the quality of the interpolation can be formed using radial basis functions:

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

where $\mathbf{v}_t$ is the test view, $\mathbf{v}_i$ – the stored example views, and the radial basis function is taken to be a Gaussian (here and below, both the subscript A indicating the object, and the superscript (p) indicating membership in the proximal view space, are omitted). The coefficients c_i are learned from examples, essentially by requiring $RBF(\mathbf{v}_i)$ to be close to 1 (the training procedure is somewhat more complicated if criteria for several objects are to be learned simultaneously, as described in section 3.1). Subsequently, $\mathbf{v}_t$ is recognized as belonging to the same object as $\{\mathbf{v}_i\}$ if $RBF(\mathbf{v}_t)$ is sufficiently close to 1. Given enough example views, the value of the recognition criterion 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.

Regardless of the algorithmic details, the tacit assumption in the recognition scenario such as the one sketched above 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 RBF 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

Standard approaches to recognition (including the multiple-view scheme just described) tend to share several common features, stemming from the assumption that variability in object appearance is mainly due to extrinsic factors such as illumination and pose. First, recognition schemes strive to

separate the intrinsic shape of the viewed object from the influence of these factors. Second, they tend to rely on geometric representations of familiar objects, the more detailed the better. Third, they base the recognition decision on the nearest-neighbor criterion.

A reflection on the nature of everyday recognition tasks prompts one to question the validity of the last two components of this strategy. Human observers, for example, are expected to ignore much of the shape details in a normal *categorization* situation (Rosch, 1978; Price and Humphreys, 1989; Smith, 1990). Barring special (albeit behaviorally important) cases such as face recognition, entry-level (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 Dimensionality reduction and the proximal 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 fiducial sample points (see Figure 1). For example, according to a definition stated and developed in (Kendall, 1984), a family of shapes each of which is a "cloud" of k points in R^3 spans a $3k$-dimensional *shape space* $\sum_3^k$ (cf. the notion of a 3D-view of an object, mentioned above). Moving the k points around in R^3 (or, equivalently, moving around the single point in the

shape space $\sum_3^k$) amounts to changing ("morphing") one shape into another.

By defining similarity between shapes via a distance function in a 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.

The proximal measurement space $\mathcal{P}$ in which the estimation of distances and the clustering are actually performed is typically very high-dimensional, for good reasons. 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. Most of this high-dimensional space is, however, 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). Together, these factors define a *proximal object space* $\mathcal{O} \subset \mathcal{P}$. This space possesses a low-dimensional structure, which can be put to use both for recognition and for categorization. On the one hand, rotating the object in depth (a transformation with two degrees of freedom) gives rise to a two-dimensional manifold which we called the view space of the object, $\mathcal{V} \subset \mathcal{O}$. On the other hand, smoothly changing the shape of the imaged object causes its measurement-space representation to ascribe a manifold $\mathcal{S} \subset \mathcal{O}$, whose dimensionality depends on the number of degrees of freedom of the shape change (for example, simple morphing of one shape into another produces a one-dimensional manifold).

In this formulation, the categorization problem becomes equivalent to determining the location of the image (i.e., the measurement-space representation) of the input object within $\mathcal{S}$. Our approach to this problem is inspired by the observation that the location of a point on a manifold (e.g., in a terrain) can be precisely defined by specifying its distance to some prominent reference points, or *landmarks* (Edelman and Duvdevani-Bar, 1997b). The estimation of the distance here (carried out perforce in the measurement space $\mathcal{P}$) must exclude "irrelevant" components that are orthogonal to the entire object space $\mathcal{O}$ (a simple example of such a component is the difference between the mean intensity levels of two images). Moreover, because our distance is meant to capture difference in shape (i.e., the amount of deformation), components due to transformations such as rotation, which lie within $\mathcal{O}$ but are (locally) orthogonal to $\mathcal{S}$, must also be discounted.

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 $\mathcal{P}$ to a small number, equal to the number of modules (Figure 2). In the next section, we describe a system for shape categorization implementing this approach, which we call the Chorus of Prototypes (Edelman, 1995).

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 space $\mathcal{O}$. 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 members of $\mathcal{O}$. 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 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, which we assume here, for simplicity, to be radial (that is, to depend only on the distance between the actual input and the original data point, which serves as its center; ideally, the shape of the basis function should reflect the statistics of the input space). The resulting scheme is known as radial basis function (RBF) interpolation.

A simple version of the RBF model of object recognition which we already mentioned above (eq. 2) uses one basis function for each familiar view $\mathbf{v}_i$. The appropriate weight c_i for each basis can be computed in this case by straightforward matrix inversion. To determine whether a test view $\mathbf{v}_t$ belongs to the object on which the network has been trained, it 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 multiplied by the weights c_i to determine the output of the network, as described by eq. 2, and illustrated in Figure 3, inset.

3.2 Multi-classifier system design

Our approach to recognition and categorization calls for several modules, each tuned to a different shape class. Before training the modules, the system selects a small subset of the familiar views of each object. These are chosen so as to optimize a "canonical distortion" criterion, which is a kind of vector quantization error (Baxter, 1997). The algorithm employed for this purpose determines both the training views (i.e., the locations of the basis functions) and the associated Gaussian spread constants, while attempting to optimize performance across the entire collection of RBF modules, rather than for each module separately. The optimization criteria require that the clusters corresponding to the views of the different objects in the space of module outputs be both tight and well-separated (see appendices B and C). Having chosen the ensembles of training views and the spread parameters by canonical vector quantization, our system proceeds to train the individual RBF modules, as it is done in the simple case described above.

The response properties of the multi-module ensemble are illustrated in Figure 16, which shows the activity of several RBF units 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.[3] In the next section we describe a series of computational experiments that examine the representational capabilities of a multi-classifier system in a range of tasks.

[3]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.

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-module Chorus system (see Figure 5).

The focus of the present study is on shape-based recognition, or, more generally, on problems arising from the processing of information contained in familiar and novel shapes. Consequently, issues such as figure-ground segmentation, as well as illumination, translation and scale invariance, were excluded from consideration. All 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).

The performance of the 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. For each module, about a tenth of the 169 test views of the corresponding object, determined by canonical vector quantization (see appendix B.1), had been used for training. The majority of the test views were therefore novel (to the RBF module, albeit not to the vector quantizer); the orientation of the object in many of these differed by tens of degrees from the closest familiar view.

	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%.

4.1.1 Identification results

We first computed the performance of each of the ten RBF modules using individually determined thresholds. The threshold of each module was set to the mean activity on the trained views, less one standard deviation. The performance of the ten modules on their training objects is summarized in Table 1. The residual error rates were about 10%, a figure that can probably be improved if a better image transduction stage, a more robust view interpolation method and a more thorough 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 produces 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%.

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. As expected, the WTA algorithm applied to the second-level RBF output resulted in a lower error rate: 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.

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 reference (trained) 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 formed 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 this stage. In every case, the performance of the 10-dimensional Chorus-based representation was compared to that of the original 200-dimensional receptive-field (RF) measurement space (see Figure 4).

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 the 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. One may observe 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 the validity of this assumption for human vision in a psychophysical experiment (Duvdevani-Bar, 1997), in which subjects were required to judge similarity among the same shapes used in the present study, on the basis of shape cues only. Similarity scores[4] 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]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.

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 was made to store N views of each reference object, each represented as a point in the 10-dimensional space of module outputs ($10N$ numbers 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 sets of labels, 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, they are well-approximated by their centroids. 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 rates of 23%, 16% and 14% for the three sets of category labels (with the average taken over values of k ranging from 1 to 9). In the measurement space, the misclassification rates were higher; on the 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 $k = 3$.

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.

4.2.5 Lessons from the categorization experiments

The pattern of performance of the various algorithms tested in the categorization tasks conformed to the expectations. In particular, the 10-dimensional representation space spanned by the outputs of the RBF modules was better than the "raw" 200-dimensional measurement space. Despite those encouraging results, the best performance of the system in the categorization experiments (about 85% correct rate) falls short of the nearly perfect 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`).[5]

As in Figure 7, our system clustered views by object identity, and grouped view clusters by similarity between the corresponding objects. To obtain a quantitative estimate of the discrimination performance afforded by this representation, we used the k-NN algorithm, as explained in section 4.2.2, this time with labels corresponding 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 $\pm 20^\circ$ around the canonical orientation of each test object were considered, the mean 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, the same experiment yielded an 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, the error rate was on the average higher, presumably 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

[5] A more rigorous treatment of the issue of informativeness of reference objects could be based on the Bayesian framework.

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 may stem 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 in other categorization tasks, an observation that is supported by our data.

To quantify the ability of the system 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 system's performance with the number of test objects seems to be satisfactory.

5 Discussion

The main goal of the present work had been to develop an approach to object representation that would be able to deal with novel shapes, subject to a practical constraint: it had to rely on view interpolation, because of this method's learnability (Edelman, 1993) and its demonstrated effectiveness in achieving nearly viewpoint-invariant performance with a minimal computational effort (Poggio and Edelman, 1990). Another design constraint that we employed, whose full treatment is beyond the scope of the present paper, stressed the desirability of a rigorous link between the internal representation of an object and its geometrical form (Edelman and Duvdevani-Bar, 1997b). This constraint would be satisfied trivially by a representation that is a geometrical replica of its target. More to the point, it can also be satisfied by ensuring that the mapping between the object's geometry and its "location" in the internal representation space is smooth (in the terminology of section 2, the representation space is the shape space $\mathcal{S}$). Indeed, the mapping in question *is* smooth in a system comprised of several view interpolation modules, which represent an object jointly, by the vector of its similarities to the several reference shapes.

The operation of an individual module in our system is best understood in the wider context of

Repr. Space	Category labeling	Method				
		WTA		k-NN M	1-NN	k-NN C
		1st	2nd			
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. Under all methods, the errors improved when the second and the third sets of category labels were used.

a family of algorithms that exploit geometric constraints inherent in a pre-established correspondence between features in two or more views of the same rigid object (Ullman and Basri, 1991; Vetter and Poggio, 1997). The same geometric constraints are also used in the varieties of viewpoint normalization and alignment (Lowe, 1987; Ullman, 1989; Breuel, 1992; Tomasi and Kanade, 1992). All these view-based methods, which require the knowledge of image coordinates of fiducial points or other geometric primitives in each of the input views, are frequently contrasted with appearance-based methods, which use collections of images without explicit correspondence. The distinction between view- and appearance-based methods becomes, however, blurred when objects are considered in isolation, following segmentation from the background and "rough alignment" (Ullman, 1996) such as centering and size normalization. In such a case, a measurement vector produced by correlating the image ("appearance") of an object with filters anchored to particular image locations contains (albeit implicitly) information about the coordinates of geometry-related features (e.g., image edges). Thus, the recognition of an individual object can be carried out by interpolation in the shape "appearance" space $\mathcal{S}$, by invoking the same principle that makes this possible in the space of the coordinates of geometric features.

It has not been easy to adapt any of the recognition methods to carry out categorization. Even the most successful recognition systems tend to ignore the challenge posed by the problems of representation and of categorization of novel objects (Murase and Nayar, 1995), or treat categorization as a kind of imprecise recognition (Basri, 1996), which may be achieved, as it were, as a byproduct of the "real" recognition algorithm (Mel, 1997; Nelson and Selinger, 1998). The main difficulty facing the extension of recognition algorithms to categorization stems from the excessive amount of geometric detail in pictures: much of the information in a snapshot of an object is unnecessary for its categorization, as attested by the ability of human observers to classify rough line drawings of common shapes (Biederman and Ju, 1988; Price and Humphreys, 1989). While image metrics that downplay within-category differences could be defined in some domains (such as the classification of stylized "clip art" drawings; see Ullman, 1996, p.173), more general attempts to classify 3D objects (vehicles) by alignment to geometric models met with only a limited success (Shapira and Ullman, 1991).

We believe that the extension of recognition algorithms (in particular, alignment) to categorization is problematic for a deeper reason than mere excess of information in images of objects. Taking recognition by alignment as an example, we may note that both stages in this process (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 Chorus scheme, on which our present approach is based (Edelman, 1995; Edelman, 1998), 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. The minimalistic implementation of Chorus described in the preceding sections achieved shape-based recognition and generalization performance approaching that of the best recognition systems (Murase and Nayar, 1995; Mel, 1997; Schiele and Crowley, 1996; Colin de Verdiére and Crowley, 1998). More importantly, our model also exhibited significant capabilities for shape-based categorization and for useful representation of novel objects.

The ability to make sense of novel objects has been considered traditionally as a prerogative of structural methods (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 problem of novelty is addressed there 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.

Recently, a number of research groups have been attempting to combine the simplicity of appearance-based features (such as correlations with Gabor patches or even image snapshots) with the robustness of structural descriptions. The most promising approaches involve the estimation of 2D, image-based feature layout (as opposed to 3D, object-centered structure). In one example, evidence concerning object identity is iteratively refined by considering mutual constraints based on relative locations of simple template-like features in an image (Amit and Geman, 1997). In another example, image-based structural relations among "Gabor probes" are used to recognize several classes of shapes such as chairs, benches and tables (Burge et al., 1997). In contrast to this basically structural approach, Burl et al. (1998) invoke a geometric observation that can be traced back to Lowe's (1986) viewpoint consistency constraint: the positions of a few features in the image constrain the positions of other features of the same rigid object. The difference between their approach and the varieties of alignment developed in the 1980's lies in a newly forged link to the statistical theory of shape (Kendall, 1984): the verification step in the new algorithms involves decision-making on the basis of distributions of likely feature locations, rather than an unprincipled application of some feature-space metric.

The method described in (Burl et al., 1998) has been tested mainly on face detection in cluttered

scenes and on cursive character recognition. There is evidence that an empirical approach based on similar principles — encoding rough structure of objects in image coordinates — can also support general object recognition in the presence of occlusion and clutter (Nelson and Selinger, 1998). Importantly, the system of Nelson and Selinger can perform categorization of novel instances of familiar classes. It also represents object structure explicitly, making it, in principle, capable of reasoning about object parts — a serious challenge for a method such as ours, which is based on unstructured appearance data.

In summary, although it seems that research in high-level vision generally concentrates upon recognition at the expense of categorization, considerable progress has been made recently in addressing both these problems. Our contribution to this progress is a simple computational scheme that can perform both recognition and categorization, by encoding any input by its similarity to a number of reference shapes, themselves represented by specially trained dedicated modules. The performance of our system (see Table 3) and of its derivatives (Edelman and Duvdevani-Bar, 1997a; Duvdevani-Bar et al., 1998) suggests that this principle may allow for efficient representation, and, in most cases, correct categorization, of shapes never before encountered by the observer. Our system has also a number of severe limitations: (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 appearance-based schemes). There are reasons to believe that translation and scaling can be treated effectively without abandoning the present approach (Vetter et al., 1995; Riesenhuber and Poggio, 1998). We conjecture that it can be extended to scenes and to the explicit representation of structure, by letting each of the reference-object modules extract and process coarse channel-coded information on the image location of its target, as suggested in (Edelman, 1998).

Acknowledgments

We thank M. Dill, N. Intrator, T. Poggio and P. Sinha for helpful suggestions concerning an early version of this manuscript, and the anonymous reviewers for constructive comments.

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*, 19:721–732.

Amit, Y. and Geman, D. (1997). Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588.

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

Baxter, J. (1997). The canonical distortion measure for vector quantization and function approximation. In D. H. Fisher, J., editor, *Proc. 14th Intl. Conf. on Machine Learning*, pages 39–47, Nashville, TN.

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.

Burge, M., Burger, W., and Mayr, W. (1997). Recognition and learning with polymorphic structural components. *Journal of Computing and information Technology*, 4:39–51.

Burl, M., Weber, M., Leung, T., and Perona, P. (1998). *From Segmentation to Interpretation and Back: Mathematical Methods in Computer Vision*, chapter "Recognition of Visual Object Classes". Springer-Verlag. in press.

Colin de Verdiére, V. and Crowley, J. L. (1998). Visual recognition using local appearance. In *Proc. 4^{th} Europ. Conf. Comput. Vision, H. Burkhardt and B. Neumann (Eds.), LNCS-Series Vol. 1406–1407, Springer-Verlag*, volume 1, pages 640–654.

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

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., Edelman, S., Howell, A. J., and Buxton, H. (1998). A similarity-based method for the generalization of face recognition over pose and expression. In Akamatsu, S. and Mase, K., editors, *Proc. 3rd Intl. Symposium on Face and Gesture Recognition (FG98)*, pages 118–123, Washington, DC. IEEE.

Edelman, S. (1993). On learning to recognize 3D objects from examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:833–837.

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

Edelman, S. (1998). Representation is representation of similarity. *Behavioral and Brain Sciences*, 21:449–498.

Edelman, S. and Duvdevani-Bar, S. (1997a). Similarity-based viewspace interpolation and the categorization of 3D objects. In *Proc. Similarity and Categorization Workshop*, pages 75–81, Dept. of AI, University of Edinburgh.

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

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.

Lowe, D. G. (1986). *Perceptual organization and visual recognition.* Kluwer Academic Publishers, Boston, MA.

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.

Nelson, R. C. and Selinger, A. (1998). Large-scale tests of a keyed, appearance-based 3-D object recognition system. *Vision Research*, 38:2469–2488.

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. 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 Vetter, T. (1992). Recognition and structure from one 2D model view: observations on prototypes, object classes, and symmetries. A.I. Memo No. 1347, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

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 Jordan, M. I., Kearns, M. J., 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.

SAS (1989). *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.

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.

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. and Poggio, T. (1997). Linear object classes and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:733–742.

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 examine the theoretical underpinnings of the ability of an RBF module to overcome the effects of pose changes. Suppose that an RBF module has been trained on a set of object views $V = \{\mathbf{x}_i\}$. We show that the module's response to a view that differs from that set by a small displacement that leaves it *within* V is always higher than the response to views obtained by a displacement directed away from V.

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 $\Delta\mathbf{x}$.

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:

$$\begin{aligned} RBF(\mathbf{x}) &= \sum_{i=1}^{k} c_i \, G(\|\mathbf{x} - \mathbf{x}_i\|) \\ &= \sum_{i=1}^{k} c_i \, e^{-[(\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}-\mathbf{x}_i)]^2/\sigma^2}. \end{aligned} \tag{3}$$

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:

$$\mathbf{c} = \mathbf{G}^{+}(\mathbf{X};\mathbf{X}) \cdot \mathbf{y}, \tag{4}$$

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

Thus, equation (3) takes the form

$$RBF(\mathbf{x}) = \mathbf{G}(\mathbf{x};\mathbf{X}) \cdot \mathbf{G}^{+}(\mathbf{X};\mathbf{X}) \cdot \mathbf{y}. \tag{5}$$

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

$$\begin{aligned} &\frac{\partial RBF(\mathbf{x} + t\Delta\mathbf{x})}{\partial t} \Big|_{\substack{\mathbf{x}=\mathbf{x}_1 \\ t>0,\ t\to 0}} = \\ &\frac{\partial}{\partial t}\left[\sum_{i=1}^{k} c_i \; e^{-[(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)]^2/\sigma^2}\right] = \\ &\sum_{i=1}^{k} c_i \; e^{-[(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}_1+t\Delta\mathbf{x}-\mathbf{x}_i)]^2/\sigma^2}. \\ &\quad \cdot \tfrac{\partial}{\partial t}\{-[(\mathbf{x}_1 + t\Delta\mathbf{x} - \mathbf{x}_i)^T(\mathbf{x}_1 + t\Delta\mathbf{x} - \mathbf{x}_i)]^2/\sigma^2\}. \end{aligned} \tag{6}$$

Denote

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

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

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

and,

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

Thus,

$$D = -\frac{2}{\sigma^2}\left[\|\mathbf{x}_1 - \mathbf{x}_i\|^2 + 2t(\Delta\mathbf{x})^T(\mathbf{x}_1 - \mathbf{x}_i)\right]\left[2(\Delta\mathbf{x})^T(\mathbf{x}_1 - \mathbf{x}_i) + 2t\right]. \tag{7}$$

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 $\Delta\mathbf{x}$ leaves $\mathbf{x}$ within the set of views $\mathbf{x}_i$, $i = 1, \ldots k$, whereas in case (A), the direction of the displacement is away from that set (see, again, Figure 11). Denote, $d_i \stackrel{\triangle}{=} \|\mathbf{x}_1 - \mathbf{x}_i\|$, $\Delta_i \stackrel{\triangle}{=} (\Delta\mathbf{x})^T(\mathbf{x}_1 - \mathbf{x}_i)$, and note that $d_i \geq 0$. With the new notation, equation (7) 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 (7) we have,

$$\frac{\partial RBF(\mathbf{x} + t\Delta\mathbf{x})}{\partial t} \mid_{\substack{\mathbf{x}=\mathbf{x}_1 \\ t>0,\, t\to 0}} \underset{t\to 0}{\longrightarrow} \sum_{i=1}^{k} c_i\, e^{-\frac{{d_i}^2}{{\Delta_i}^2}} \cdot (-\frac{4}{\sigma^2}d_i\Delta_i). \tag{8}$$

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 within the set of views results in a smaller change of the corresponding RBF activity

than the RBF change resulted from a displacement that is away from it. 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 (4) 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 $d_{ij} \stackrel{\triangle}{=} \|\mathbf{x}_i - \mathbf{x}_j\|$. Note that $\mathbf{G} = I + A$, where I is a unit matrix[6], 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,

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

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

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

(object rotation), and (b) an intrinsic shape change, that is, a change 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

$$G(\|\mathbf{x}-\mathbf{t}\|) = e^{\frac{-\|\mathbf{x}-\mathbf{t}\|^2}{\sigma^2}} = e^{\frac{-\|\mathbf{x}^p-\mathbf{t}^p\|^2}{\sigma^2}} e^{\frac{-\|\mathbf{x}^v-\mathbf{t}^v\|^2}{\sigma^2}}. \tag{9}$$

Consider now a displacement along an object's 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 $\mathcal{X}$, 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,

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

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,

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

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,

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

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

Hence,

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

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 $\sigma < 2 \min_{\substack{i,j \\ i<j}} d_{ij}$, for $d_{ij} \stackrel{\triangle}{=} \|\mathbf{x}_i - \mathbf{x}_j\|$.
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

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

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 (9)),

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

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

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-space 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 along 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; neither the number nor the profile of these receptive fields proved critical in a preliminary investigation). 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, 1997; 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 (1997). 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.[7]

[7]Formally, for an environment of functions $f \in \mathcal{F}$, mapping a probability space (X, P, σ_X) into a space (Y, σ), with

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).

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 two randomly chosen 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).

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.

$\sigma : Y \times Y \rightarrow R$, 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 $\mathcal{F}$.

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*,
$Q : \mathcal{S} \rightarrow \mathcal{C}$, $\mathcal{C} = (p_1, p_2, \ldots, p_n), p_i \in \mathcal{S}, i = 1, 2, \ldots, n$. 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 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 $\mathcal{C}_m^N$, perform the *modified Lloyd Iteration* on the classifier output space to generate the improved codebook $\mathcal{C}_{m+1}^N$ (i.e., the set of *input* vectors, whose classifier *outputs* are the closest to the code-vectors constituting the improved output codebook). The modified Lloyd Iteration is defined as follows:

i. Augment the codebook $\mathcal{C}_m^N$ by a random member of the input vector set.

ii. Train the classifiers and compute their activities over the input set; denote this set by $\mathcal{O}$. Denote the set of classifier outputs on the codebook $\mathcal{C}_m^N$ the *output codebook*.

iii. Partition the set $\mathcal{O}$ into clusters using the *Nearest Neighbor Condition*, for the output-codebook vectors being the cluster centers.

iv. Using the *Centroid Condition*, compute the centroids for the clusters just found, to obtain a new output codebook.

(c) Compute the canonical distortion D_m (see section B.1).
If there is no improvement, mark and discard the latest addition to the codebook; go to Step (b). If $\frac{D_m - D_{m+1}}{D_m} < \epsilon$ for a suitable threshold ϵ, continue to Step (3).
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).

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: *(see preceding page)* 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. The 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).

	♯ Reference Objs				
♯ Test 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).

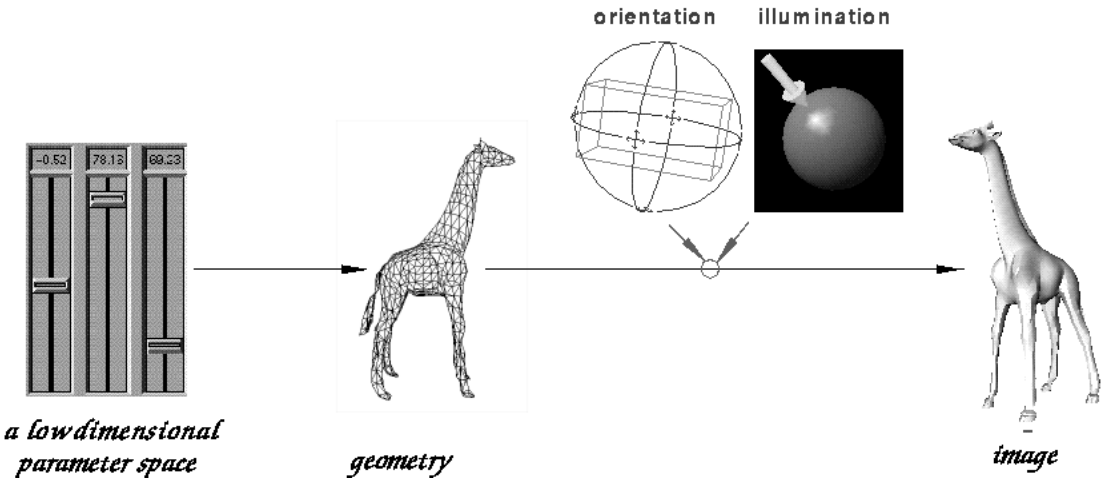

Figure 1: The process of image formation. A family of shapes (say, 4-legged animal-like objects) can be given a common parameterization in terms of the the locations of the vertices of a triangular mesh that approximates the object's shape. For sufficiently similar shapes, the parameterization is effectively low-dimensional, as illustrated symbolically on the left by the three "sliders." Intrinsic and extrinsic factors (shape and viewing conditions) together determine the appearance of the object.

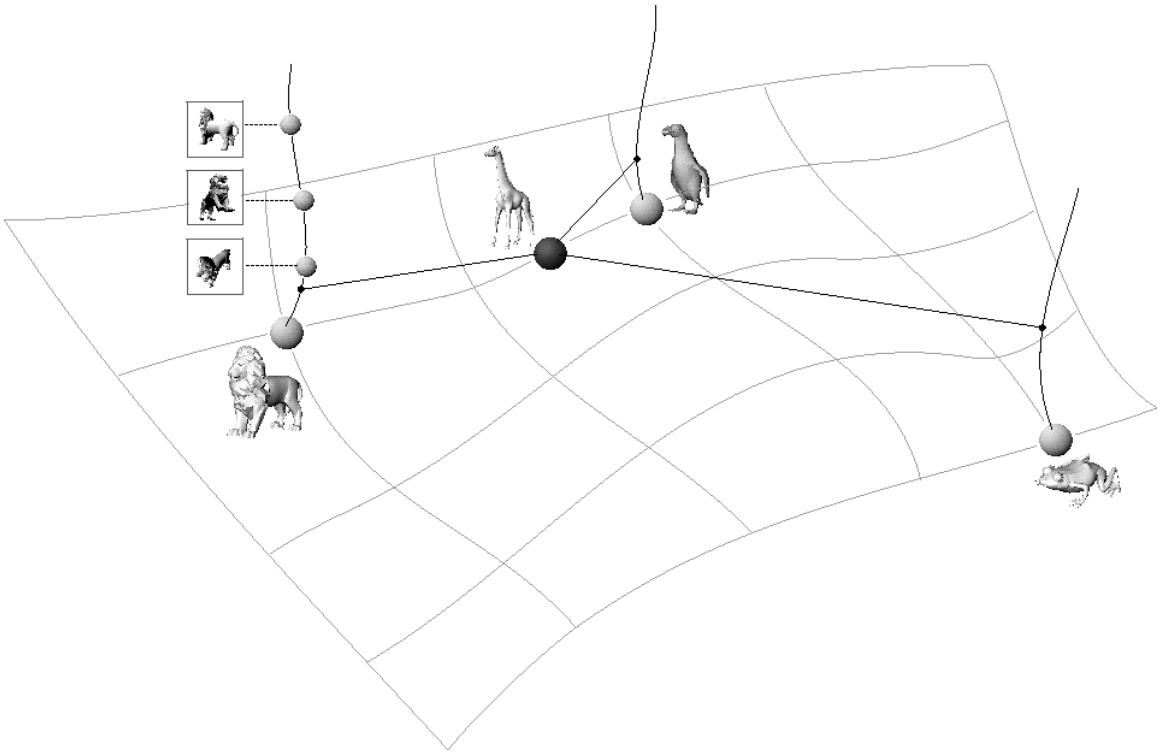

Figure 2: A schematic illustration of the shape-space manifold defined by a Chorus of three active modules (`lion`, `penguin`, `frog`). Each of the three reference-shape modules is trained to ignore the viewpoint-related factors (the view space dimension, spanned by views that are shown explicitly for `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 an be lower than its nominal dimensionality (equal to the number of modules); here the space is shown as a two-dimensional manifold.

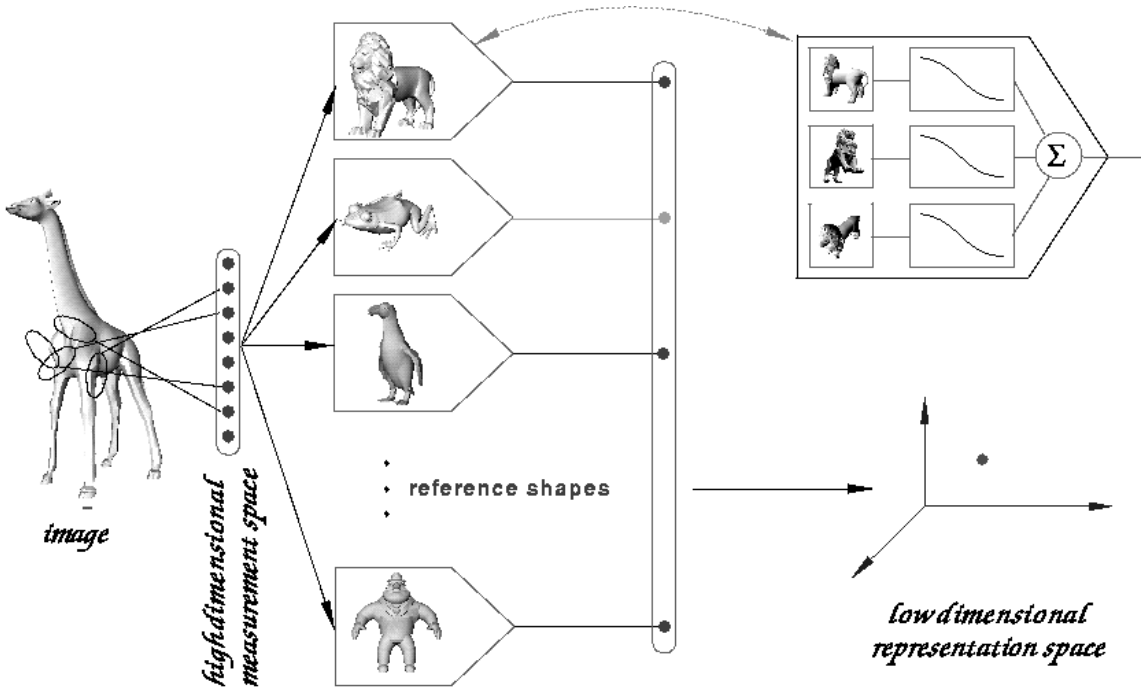

Figure 3: The Chorus scheme (section 3). The stimulus is first projected into a high-dimensional measurement space, spanned by a bank of receptive fields. Second, it is represented by its similarities to reference shapes. In this illustration, only three modules respond significantly, spanning a shape 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 $y = \sum_t c_t \mathcal{G}(\mathbf{x}; \mathbf{v}_t)$. The weights c_t and the spread parameter σ are learned as described in 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

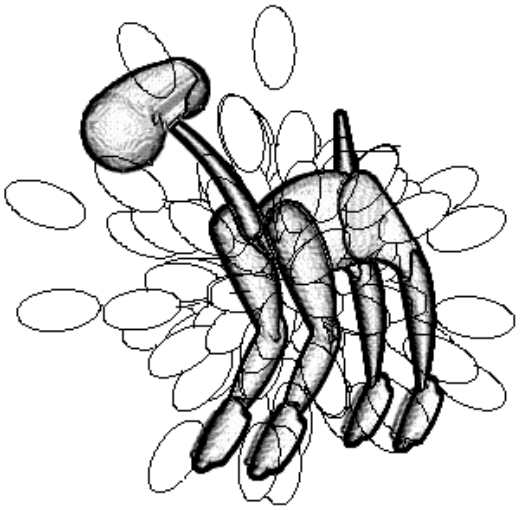

Figure 4: An image of a 3D object, overlayed by the outlines of the receptive fields (RFs) used to map object views into a high-dimensional measurement space (see appendix B). The system described here involved 200 radially elongated Gaussian RFs; only some of them are drawn in this figure.

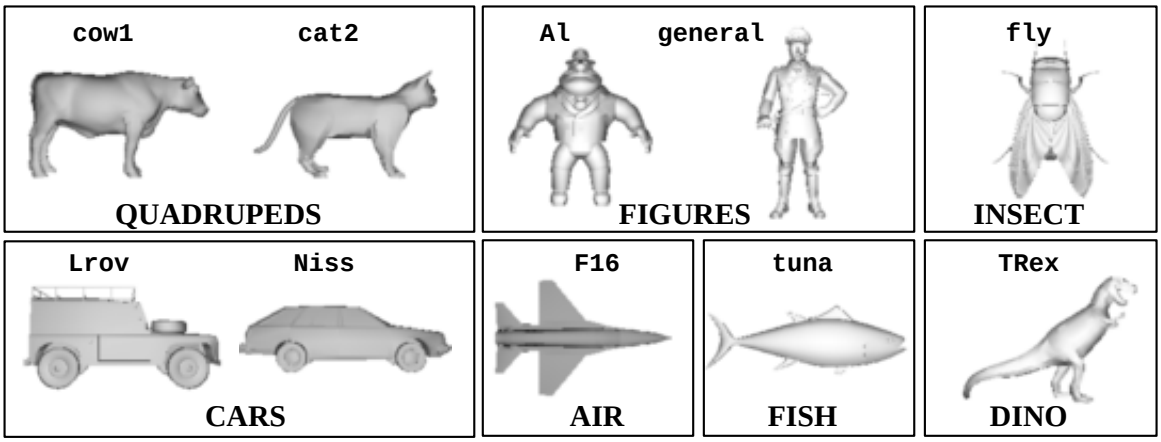

Figure 5: The ten training objects used as reference shapes in the computational experiments described in the text, organized by object categories. The objects were chosen at random from a collection available from Viewpoint Datalabs, Inc. (http://www.viewpoint.com/).

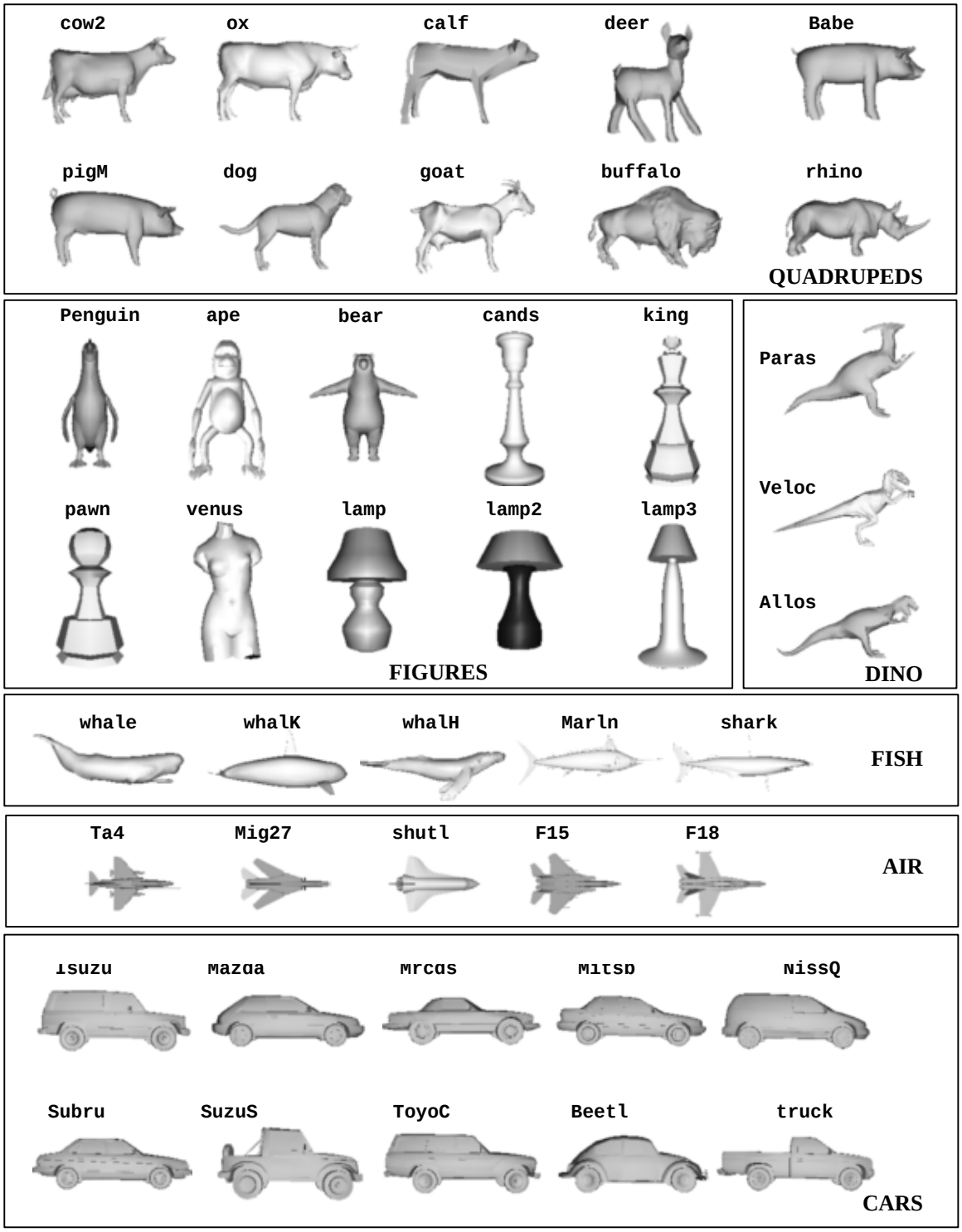

Figure 6: The 43 novel objects used to test the categorization ability of the model (see section 4.2); objects are grouped by shape category.

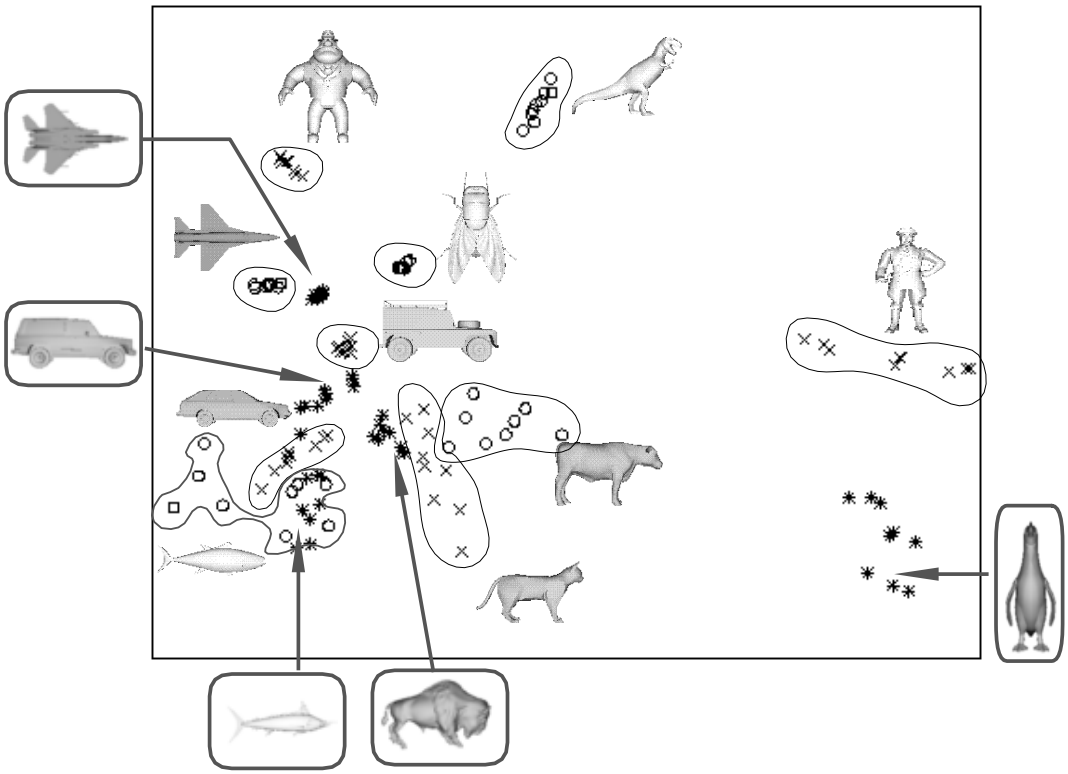

Figure 7: A 2D plot of the 10-dimensional shape space spanned by the outputs of the RBF modules; multidimensional scaling (MDS) was used to render the 10D space in 2D, while preserving as much as possible distances in the original space Each point corresponds to a test view of one of the objects; nine views (spanning the entire $60° \times 60°$ range at equal intervals) are shown 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.

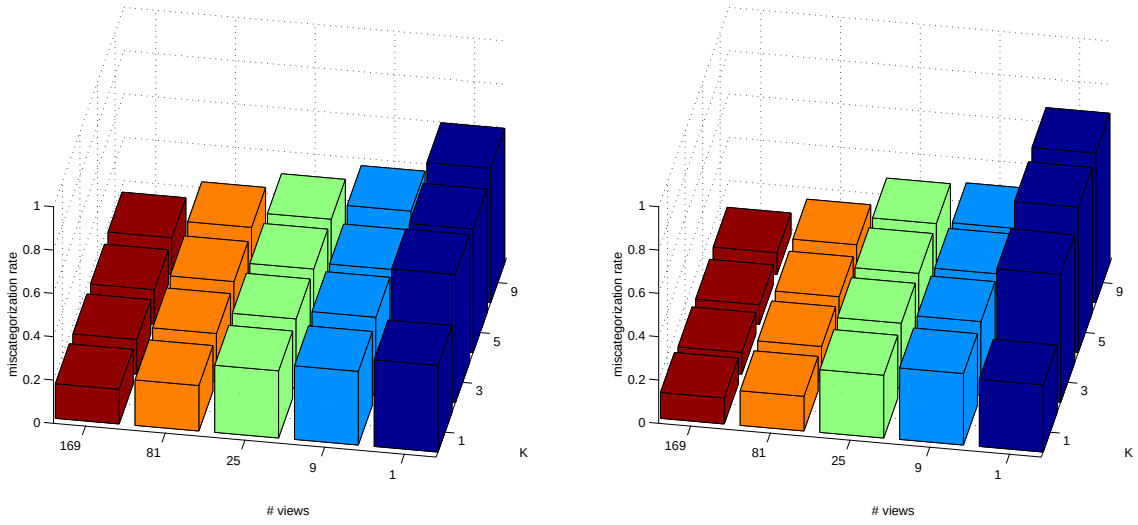

Figure 8: The performance of the k-NN procedure described in section 4.2.2 for the third set of category labels, plotted vs. k and N. The plots show the misclassification rate for the 43 test objects shown in Figure 6. *Left:* errors using the measurement-space representation; the mean misclassification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

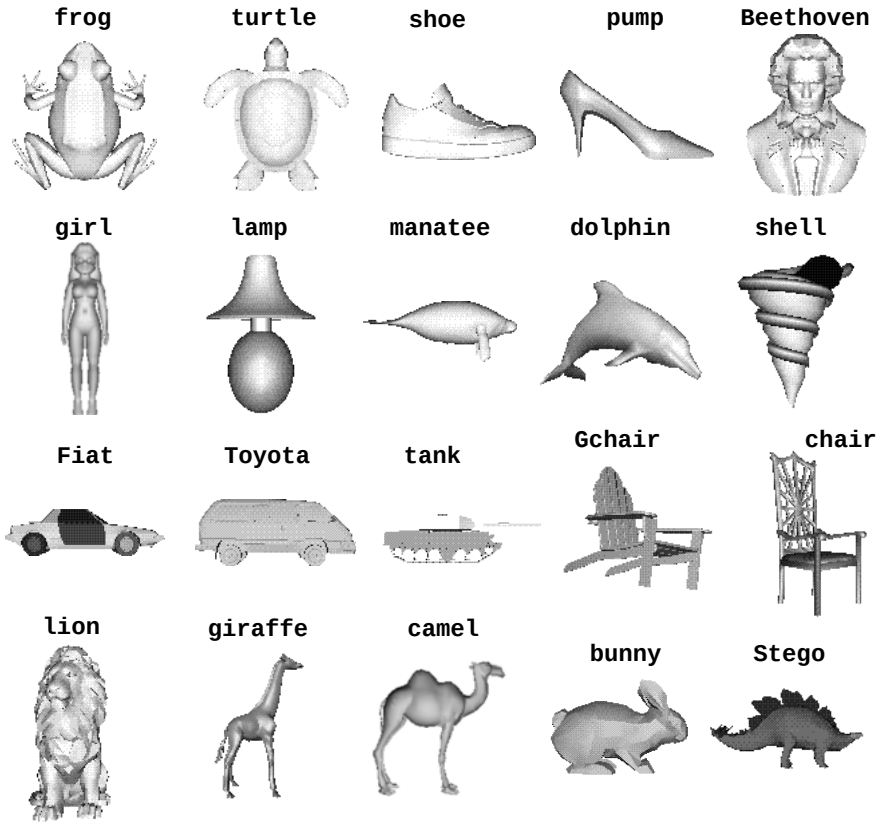

Figure 9: The 20 novel objects, picked at random from the object database, which we used to test the representational abilities of the model (see section 4.3).

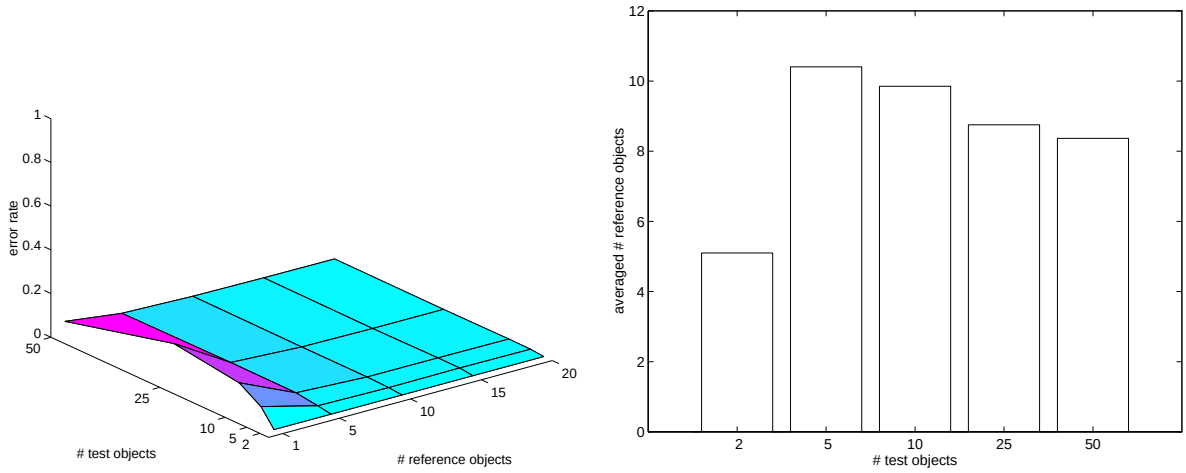

Figure 10: *Left:* the mean discrimination error rate plotted against the representation dimensionality (the number of reference objects) and the size of the test set (the number of test objects). The means were computed over 10 random choices of reference and test objects. See Table 7 in appendix D 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.

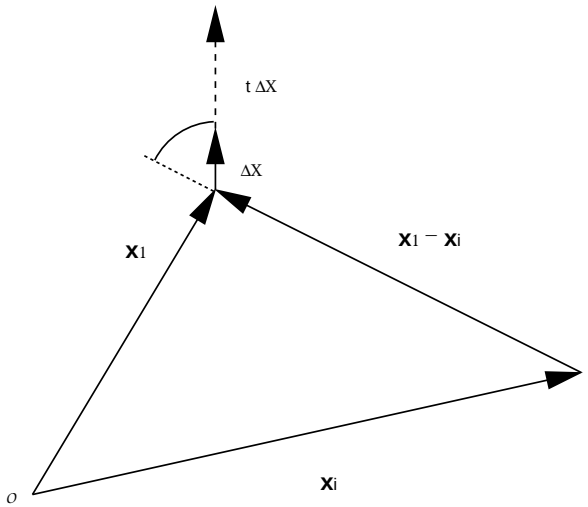

Figure 11: An illustration of the basic notation used in the text; $\mathbf{x}_1$, $\mathbf{x}_i$ are training views of a specific object shape, $i = 1, \ldots, k$. $t\Delta\mathbf{x}$ is a vector representing a displacement from the view space spanned by the training vectors. The angle between $t\Delta\mathbf{x}$ 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.

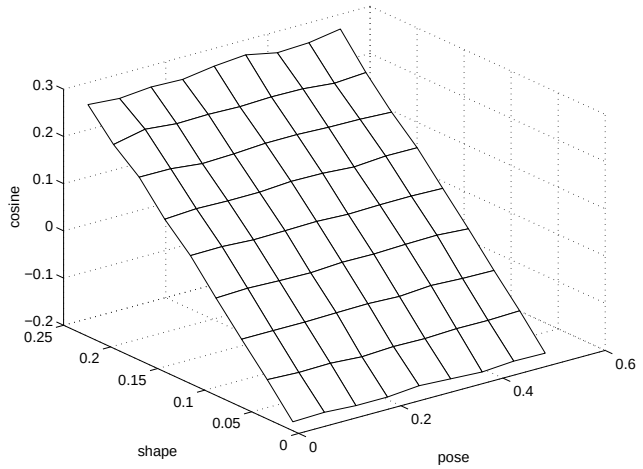

Figure 12: Orthogonality of shape and pose. The displacement in the two-dimensional appearance of a three-dimensional 10-point cloud object due to variations in pose and shape is measured, assuming orthographic projection. The plot shows the average value of the cosine of the angle between the shape 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).

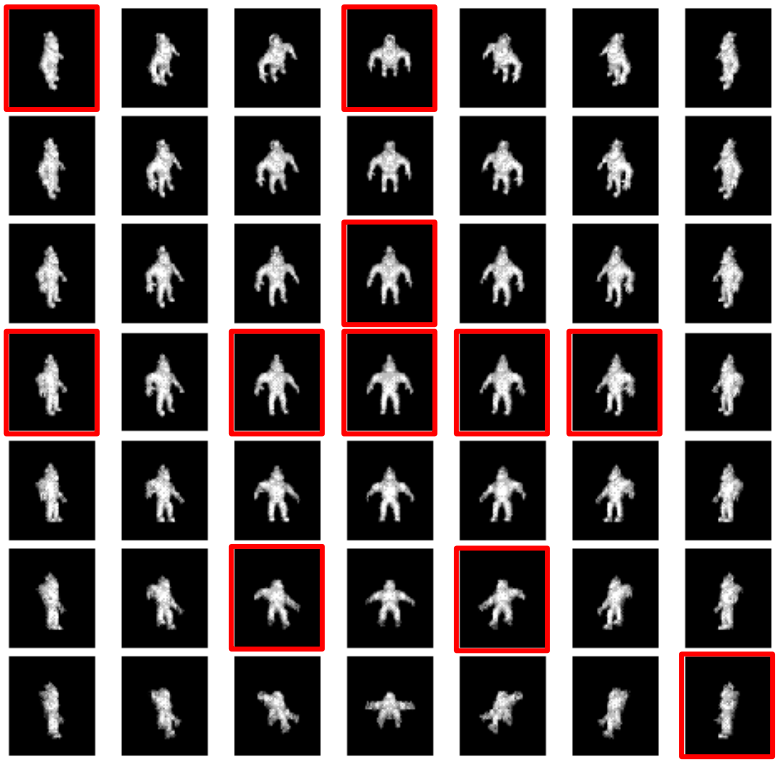

Figure 13: A set of 49 views of one of the figure-like test objects (`A1`), taken at grid points along an imaginary viewing sphere centered around the object. Views differ in the azimuth and the elevation of the camera, both ranging between $-60°$ and $60°$ at $20°$ increments. We used the Canonical 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.

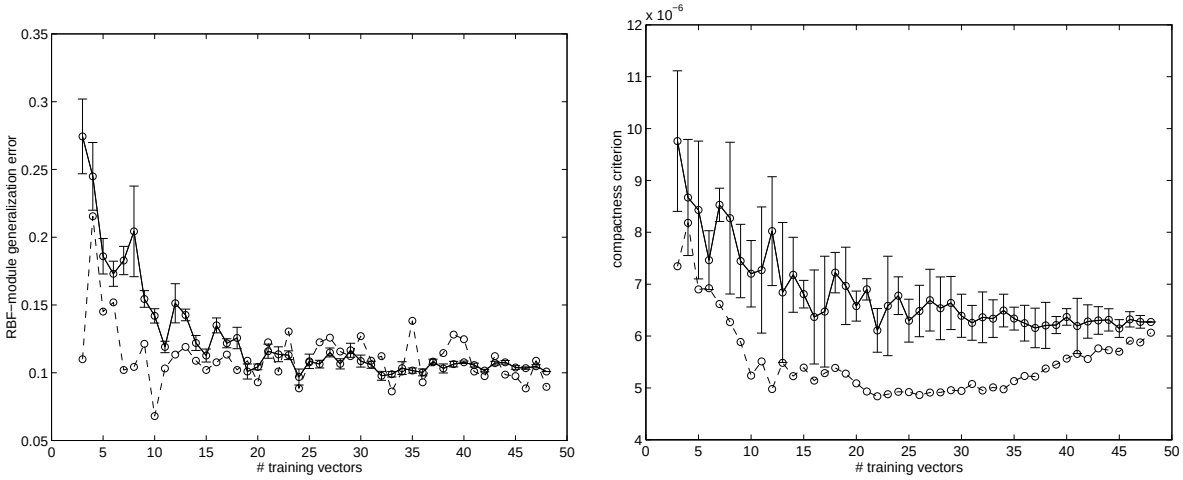

Figure 14: The effect of training-set size on the performance of an RBF module trained under the compactness criterion. *Left:* the recognition error obtained for the Nissan module, trained as a part of a network consisting of ten object modules (see Figure 5 below). For each object, training involved 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 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.

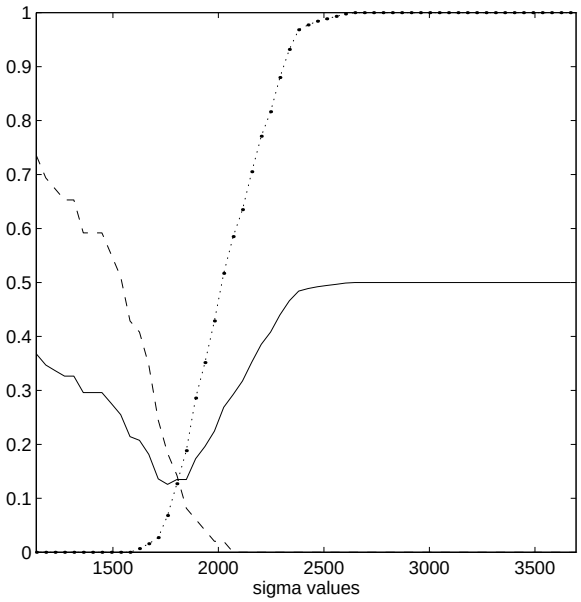

Figure 15: The effect of the basis function width (σ) on the performance of an RBF module. *Left:* RBF-module miss rate (dashed line), false-alarm rate (dotted line) and their mean (solid line), plotted against σ. The values of σ shown on the abscissa range from half the minimal distance up to the maximal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

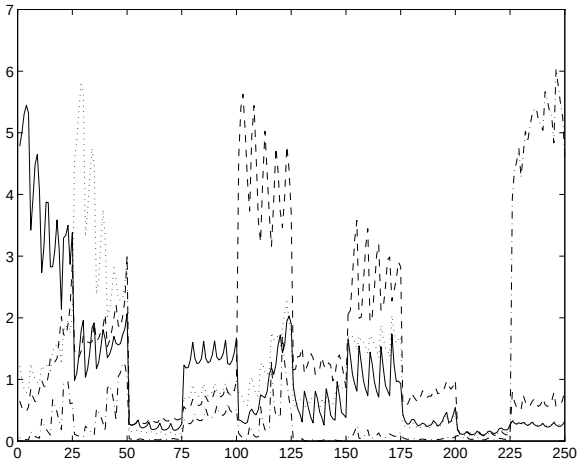

Figure 16: The activity of four RBF modules obtained for 250 test views (25 views for each of ten test objects, grouped along the abscissa). The four modules are `cow` (views 1-25, solid line), `cat` (views 26-50, dotted line), `tuna` (views 100-124, dashed line), and `TRex` (views 226-250, dash-dotted line). Note that each module responds strongly to views of its target object, and significantly less to views of other objects.

Footnotes

1. Affiliation of authors

   - Sharon Duvdevani-Bar
     Department of Applied Math,
     The Weizmann Institute of Science,
     Rehovot, 76100, Israel
     sharon@wisdom.weizmann.ac.il
   - Shimon Edelman (to whom correspondence should be addressed)
     School of Cognitive and Computing Sciences,
     University of Sussex,
     Falmer, Brighton BN1 9QH, UK
     shimone@cogs.susx.ac.uk
     Present address:
     Department of Psychology,
     Uris Hall, Cornell University,
     Ithaca, NY 14853-7601, USA

2. A more precise specification of the dimensionality and the shape of $\mathbf{v}_A$, which goes beyond the needs of the present discussion, can be found in

3. The qualifier *distal* refers to objects "out there" in the world; the space $\mathcal{V}_A$ of which $\mathbf{v}_A$ (see eq. 1) is a member is the distal view space. The qualifier *proximal*, denoted by the superscript (p), refers to entities that reside in the measurement space, such as the manifold $\mathcal{V}_A^{(p)}$.

4. 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 Typical object recognition systems in computer vision, which involve a Winner Take All decision, opt for the latter, impoverished, representation of the stimulus.

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

6. A more rigorous treatment of the issue of informativeness of reference objects could be based on the Bayesian framework.

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

8. 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 $\mathcal{F}$.

Figure Captions

1. The process of image formation. A family of shapes (say, 4-legged animal-like objects) can be given a common parameterization in terms of the the locations of the vertices of a triangular mesh that approximates the object's shape. For sufficiently similar shapes, the parameterization is effectively low-dimensional, as illustrated symbolically on the left by the three "sliders." Intrinsic and extrinsic factors (shape and viewing conditions) together determine the appearance of the object.

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 an be lower than its nominal dimensionality (equal to the number of modules); here the space is shown as a two-dimensional manifold.

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 $\mathcal{G}(\mathbf{a}, \mathbf{b}; \sigma) = exp(\|\mathbf{a} - \mathbf{b}\|^2/\sigma^2)$; 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 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

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 appendix B). The system described here involved 200 radially elongated Gaussian RFs; only some of them are drawn in this 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/).

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

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 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.

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%.

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).

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.

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.

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).

13. A set of 49 views of one of the figure-like test objects (`Al`), 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 `Al` are marked by frames.

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 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.

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.

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.

Keywords

- representation
- similarity
- visual shape recognition
- categorization
- view space
- shape space