PERGAMON

Vision Research 42 (2002) 2105–2122

Vision Research

www.elsevier.com/locate/visres

Acquisition of visual shape primitives

Ladan Shams [a,*], Christoph von der Malsburg [b,c]

[a] Division of Biology, California Institute of Technology, MC 139-74, Pasadena, CA 91125, USA
[b] Institut für Neuroinformatik, Ruhr-Universität Bochum, D-44780 Bochum, Germany
[c] Computer Science and Neuroscience Departments, University of Southern California, Los Angeles, CA 90089-2520, USA

Received 26 September 2000; received in revised form 15 April 2002

Abstract

Shape primitives have long been proposed as components for object models in the visual system, and account for a considerable body of behavioral findings. While a large amount of effort has been devoted to the study of *detection* of these parts in the scenes, no research has been undertaken simulating the acquisition of these representations. We present a model which suggests how the shape primitives may be learned by experience in a self-organized fashion. This model offers the first successful unsupervised learning of shape primitives which are as complex as object parts and can serve as intermediate representations for various objects. The algorithm uses synthetic gray-level objects, each composed of several parts (primitives or else), and shape primitives emerge as a result of partial matches between several objects. Our algorithm does not use any a priori knowledge about any attributes of the patterns to be learned; and the recurrence of these visual patterns in various objects is the only basis for their emergence as new features.

Keywords: Shape primitives; Unsupervised learning; Object representation; Object parts; Feature learning

1. Introduction

The object recognition system in humans is endowed, in its real time processing, with great robustness with respect to apparent changes in images, to a degree no computer model has yet been able to mimic even off line. It is clear that an object representation which retains the invariant information mediates this process. The nature of this invariant representation has been the subject of debate and intense study. A part-based representation in which objects are represented in terms of the constituent shape primitives and the topological relationship amongst the parts (Biederman, 1987), possesses many important invariances consistent with those of the brain. This representation accounts for a body of psychophysical data on object recognition (Bar & Biederman, 1995; Biederman, 1987; Biederman & Cooper, 1991a,b; Biederman & Gerhardstein, 1993; Cooper, 1993), and seems to characterize the object representation employed by the brain fairly well at least for a considerable class of objects. Entry-level object recognition, e.g., discrimination between a chair and a car, seems to be robust to rotation in depth even for novel objects, and this model is arguably the only one which can account for this capacity. Of course, it is likely that for the subordinate object classification (e.g., discrimination of one type of car from another) and for irregular objects, other representations and recognition mechanisms are employed. However, even for such recognition tasks, the brain would have to exploit the regularities in the varying patterns in order to achieve invariant recognition. Thus, whether the regularities are in the form of object shape primitives, or other constellations of features which may not be as verbally describable and intuitive as volumetric object parts, the recurring patterns, or feature combinations seem to be the basis for the invariant recognition. In this paper, we focus primarily on regular object shape primitives, as this representation evidently plays an important role in adult human object recognition. However, our model, as we will show later, is applicable to the general problem of learning recurring patterns, extending to irregular, and arbitrary patterns.

Despite extensive efforts focusing on the *recovery* of shape primitive-like structures (Barr, 1981; Bergevin & Levine, 1988; Boult & Gross, 1987; Brooks, 1983;

* Corresponding author. Tel.: +1-626-395-2362; fax: +1-626-844-4514.
E-mail address: ladan@caltech.edu (L. Shams).

0042-6989/02/$ - see front matter
PII: S0042-6989(02)00130-X

Dickinson, Pentland, & Rosenfeld, 1992; Ferrie, Lagarde, & White, 1993; Helmholtz, 1962; Kumar, Han, Goldgof, & Bowyer, 1995; Nevatia & Binford, 1977; Raja & Jain, 1992; Solina & Bajcsy, 1990; Terzopoulos, Witkin, & Kass, 1988; Zerroug & Nevatia, 1993, 1996), no systematic investigation of the *acquisition* of the shape primitives in the brain has been yet conducted, neither on the experimental front nor within the computational modeling field. Therefore, it is not clear whether such representations, which seem to underlie much of the adult object recognition mechanisms, are the result of a learning process or arise through maturation encoded by genes. The first step in answering this question is to investigate whether, given the biological constraints, such a learning task is computationally feasible. To this end, we designed a model which is consistent with the existing experimental data in its assumptions, uses relatively realistic images as input, and relies on biologically plausible computational methods. We found that this model *can* learn shape primitives from limited exposure to objects.

In Section 2, we will lay out the learning model by first describing the input and the representation used, and then the main mechanism operating on the input representation, followed by the learning algorithm and the results. In Section 3, we illustrate that the learning algorithm is not specific to the learning of regular shape primitives, and can be used for unsupervised learning of arbitrary complex patterns. We make this case by successful application of the learning algorithm to a drastically different set of stimuli. We will conclude by a discussion of the relationship to previous models, and contributions, and shortcomings of the presented model in Section 4.

2. Shape primitive learning model

2.1. Input

The learning model presented here uses segmented texture-free, uniform color object representations as input. We use computer generated structures to model the preprocessed input. Each object is represented by a single 2D static image as shown in Fig. 1. We use three examples of shape primitives: cones, cubes and cylinders. These three primitives are distinguished by being the only ones which take part frequently in composition

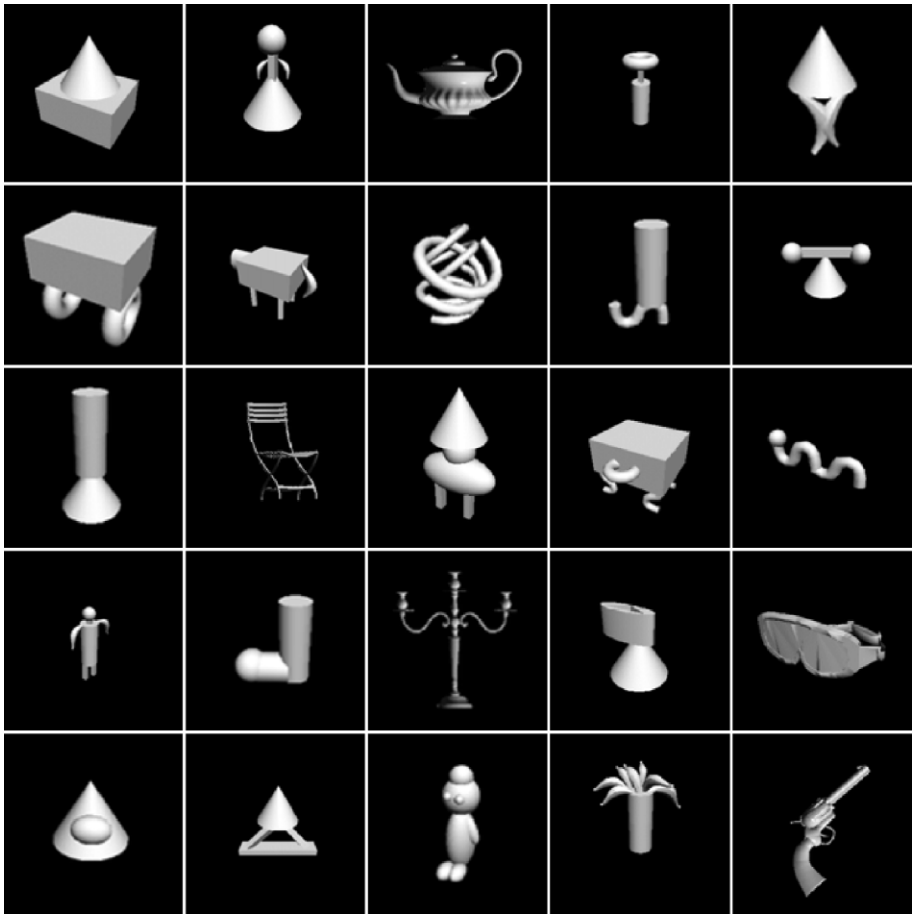

Fig. 1. Some examples of objects in the database of stored object models.

of the objects in the database. All other object parts serve as "non-primitive parts" in our model as they do not recur often in our object database. A given shape primitive appears in various objects in the database with three significant variations: size, position, and partial occlusion. That is, a given shape primitive is occluded differently in different objects, can take on varying sizes in various objects, and its position within the object varies from one object to another. For each shape primitive, we have used three different sizes, spaced about 30% apart (spanning a 60% size variation), and the three different sizes are represented with about equal frequency in the database.

The orientation in depth for a given shape primitive is not very strictly controlled from one object to another and therefore, due to this change in orientation, a given shape primitive does appear with moderate local shape deformation in different objects.

2.2. Object representation

We assume that objects are represented by a grid of hypercolumns covering the portion of the visual field segmented as the object. We model complex cell responses by the magnitude of complex-valued Gabor-wavelet responses (Lades et al., 1993). We assume a sampling of the frequency domain at three frequency levels and, within a frequency level, at four orientations. Gabor components can be expressed in terms of amplitude and phase. We refer to the amplitudes as Gabor magnitudes. We take this signal as a model of complex cell responses (Shams & von der Malsburg, in press). For a single point of the visual field we combine all such responses for different spatial frequencies and orientations into a feature vector, called "jet", which may serve as a simple model of hypercolumn activity in response to the presentation of an object image representation.

We model the array of feature cells activated by the memory trace of an object as a graph, with fixed edges of equal length to represent the geometrical lay-out in image space, and with jets attached to nodes to represent local gray-level distributions (see Fig. 2). Notice that the graph covers all of the object, and there is no distinction between different parts of the graph (e.g., the nodes falling on a shape primitive or else) in this representation.

2.3. Basic mechanism

The hypothesis underlying our model is that the brain develops shape primitive models by examining the set of stored object models and by comparing them with each other. Because shape primitives take part in the composition of the objects more frequently than any other individual structure, the statistics of their recurring projections leads to the encoding of such constellations

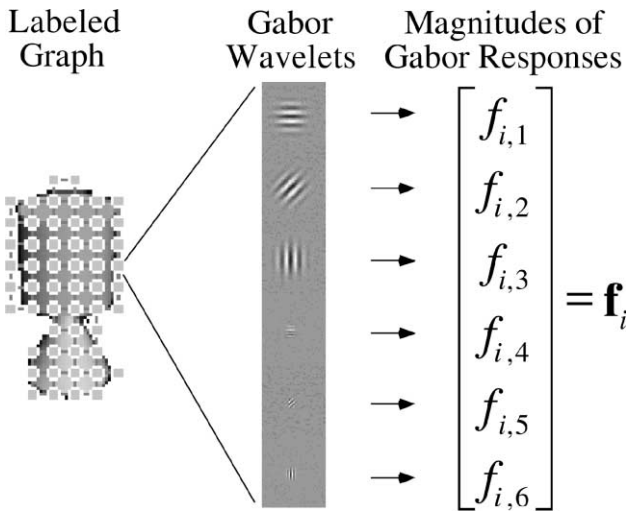

Fig. 2. The representation of an object as a labeled graph. Each object is represented by a lattice of nodes, where each node is labeled with a Gabor jet, $\mathbf{f}_i$, simulating a hypercolumn. The vector on the right is a schematic representation of a Gabor jet. Each Gabor jet is a vector of magnitudes of Gabor-wavelets responses of various orientation and frequency tunings, simulating the responses of V1 complex cells. Only two frequency levels and three orientations are shown in this diagram. Four orientations at three frequency levels were used in the model.

as new (and more complex) features. These shape primitives can then be used in the process of recognition and storage of future objects.

The findings of psychophysical studies corroborate this hypothesis (Brady, 1998; Schyns & Murphy, 1994; Schyns & Rodet, 1997). A recent study examined the learning ability of the human subjects in terms of acquiring novel patterns based on exposure to several scenes containing those patterns. It was found that subjects *were* able to learn these arbitrary (initially non-segmentable) subpatterns after exposure to several examples, i.e., several scenes containing the recurring pattern. This study clearly indicates the existence of a mechanism in the brain for actively comparing or *matching* various scenes with each other, and hence subserving extraction of recurring patterns.

We use *labeled graph matching* as the mechanism for matching different object models with each other. Below, we briefly describe labeled graph matching, and justify why we chose this method for matching.

2.3.1. Labeled graph matching

Labeled graph matching (Bienenstock & von der Malsburg, 1987; von der Malsburg, 1988) is inherently endowed with translation, scale, rotation and distortion invariance (Lades et al., 1993; Maurer & von der Malsburg, 1995; Wiskott & von der Malsburg, 1995). In our algorithm, however, we will only take advantage of translation and scale invariance, and to a small degree, distortion invariance by allowing slight local moves of the individual model nodes. The graph matching

algorithm does an exhaustive search over three dimensions: spatial position, horizontal size, and vertical size. The search in each dimension is in a subsampled space. At each spatial coordinate, the graph is resized in nine different ways, searching for the appropriate size and proportion. In each direction, horizontal or vertical, the graph can be stretched or contracted. The separateness of the horizontal and vertical dimensions in the search procedure allows for a change in proportions as well as global change in size.

2.3.2. Lateral excitation

Our version of labeled graph matching can be characterized as the positioning of one graph over another, optimizing the similarity between the matched node labels—Gabor jets—over all possible relative graph positions. The traditional measure of similarity s (Lades, 1994; Wiskott, Fellous, Krüger, & von der Malsburg, 1997) between two Gabor jets $\mathbf{f}_i$ and $\mathbf{f}_j$ is the cosine of the angle between the two jets (interpreting jets as vectors):

$$s(\mathbf{f}_i, \mathbf{f}_j) = \frac{\mathbf{f}_i^{\mathrm{T}} \mathbf{f}_j}{\|\mathbf{f}_i\| . \|\mathbf{f}_j\|} \tag{1}$$

where $\| \cdot \|$ denotes the norm. Jet normalization provides robustness to variations in contrast level.

The optimization of similarity between two graphs, G and G', with node labels $V = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n\}$ and $V' = \{\mathbf{f}'_1, \mathbf{f}'_2, \ldots, \mathbf{f}'_m\}$, respectively, requires the definition of a cost function S. A natural choice is the sum of all pairwise node label (jet) similarities:

$$S(G, G') = \frac{1}{n} \sum_{i=1}^{n} s(\mathbf{f}_i, \mathbf{f}'_{u(i)}) \tag{2}$$

where $\mathbf{f}'_{u(i)}$ is the jet to which $\mathbf{f}_i$ has been mapped, and n is the number of nodes in the graph.

Previous algorithmic implementations of graph matching for object recognition have used this cost function (Wiskott, Fellous, Krüger, & von der Malsburg, 1995; Wiskott & von der Malsburg, 1993). In the graph matching in this paper we go one step further, by using a *non-linear* sum of the similarities as the cost function. This non-linearity has a very simple form, and has a close biological analogue—lateral excitation.

As in previous implementations, the similarity between two jets is computed in our algorithm according to (1). However, we augment the graph similarity function with an element that emphasizes the topological coherence of the match. The new graph similarity function, $\widetilde{S}$, involves the enhancement of each pairwise similarity value s by its neighboring similarity values.

$$\widetilde{S}(G, G') = \frac{1}{n} \sum_{i=1}^{n} \tilde{s}(\mathbf{f}_i, \mathbf{f}'_{u(i)}) \tag{3}$$

$$\tilde{s}(\mathbf{f}_i, \mathbf{f}'_{u(i)}) = s(\mathbf{f}_i, \mathbf{f}'_{u(i)}) + s(\mathbf{f}_i, \mathbf{f}'_{u(i)}) \sum_{k} s(\mathbf{f}_k, \mathbf{f}'_{u(k)}) \tag{4}$$

where k is the index of immediate neighbors of $\mathbf{f}_i$, $\mathbf{f}'_{u(k)}$ is the jet matched with $\mathbf{f}_k$.

The structure of the similarity function $\tilde{s}$ is meant to reflect the lateral excitatory interactions among neighboring V1 hypercolumns (Gilbert, 1992). The function of this lateral excitation during graph matching is to favor matches which lead to contiguous (or topographically smooth) high-similarity profiles over matches which contain topographically isolated high-similarity values (non-smooth high value profiles) which tend to be accidental.

The motivation behind augmentation of the simple similarity measure with 'lateral excitation' was to fortify the algorithm against noise and clutter. As it is laid out in Section 2.4, our algorithm involves matching of the model graph with another graph while only part of the model graph has a corresponding match in the other

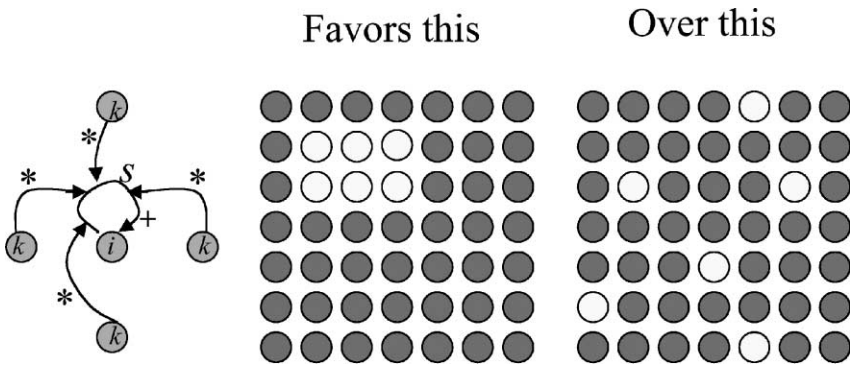

Fig. 3. The function of lateral excitation in matching. The left panel is a simplified diagram of the excitation that similarity of the node at coordinate i receives from its neighboring coordinates. The middle and right panels schematically depict two types of similarity profiles, where the light and dark circles represent high and low similarity values, respectively. The middle profile which has a more smooth configuration of high-similarity nodes is favored by lateral excitation scheme over another profile with the same number of high-similarity nodes but in dispersed spatial profile, depicted in the right panel.

graph. That is, there are parts of the model graph which do not have a corresponding match in the input graph. This is similar to the task of matching two instances of an object where the two instances are partially occluded in different ways in the two scenes. These occlusions cause the total graph similarities to drop, causing the total jet similarities of the desired match to potentially become comparable to those of false correspondences which accidentally contain some disparate high-similarity nodes. In such instances, lateral excitation favors the correct match due to its contiguity of high-similarity nodes, and in effect penalizes the false matches by suppressing the accidental isolated high-similarity nodes (see Fig. 3). This phenomenon also relates to the Gestalt principle of "continuity" or "nearness" and is also consistent with the spirit of dynamic link matching (Lades et al., 1993; Maurer & von der Malsburg, 1995; Wiskott & von der Malsburg, 1995) (a neural implementation of LGM) in that the neighboring nodes cooperate in establishing correspondences.

We have compared the matching performance between the two cost functions (2) and (3) for different model graphs against the database—104 objects, and found that the recognition performance using lateral excitation is significantly superior to the case without it. With lateral excitation, the number of correct matches is significantly larger for all numbers of false positives, and the maximum number of false positives is much smaller as well.

To verify the generality of these results, we later tested the two methods on a new set of stimuli, in a detection task. We found that the lateral excitation function outperforms the simple similarity function by large margin (Shams, Brady, & Schaal, 2001). Therefore, the addition of lateral excitation to the graph (node label) similarity computation seems to be a useful feature regardless of the specific task at hand.

2.4. Learning model

We would like to emphasize the distinction between two related but different processes. The previous studies of part-based representation have all investigated the process of *detecting* shape primitives in a scene. The focus of this paper, however, is on a different problem—that of unsupervised acquisition of the shape primitives. This problem is computationally more involved than that of shape primitive detection and recognition. While the latter tasks involve just a polynomial search, the task of learning amounts to a search for *any* possible regularity and threatens to be NP-complete. The main difficulty is that general scenes are made up of many objects arranged in an infinity of configurations and, thus, constitute a very large search space. Some of the complicating factors are background interference, inter-object occlusion; surface markings, texture, color, and varying illumination. However, the problem of learning shape primitives without any a priori knowledge, even in the absence of these complicating variations, is a quite difficult one. This can be attributed to three factors. The first is what we call *intra-object occlusion*—the partial occlusion of object parts by each other. This causes the shape primitives appearing in different objects not to look identical, due to their varying partial occlusions. A learning algorithm should, therefore, be able to extract a complete shape primitive from a set of partial or distorted examples. Another complicating factor is due to the various other parts with which a given shape primitive is combined in various objects, which act in effect as varying background for the shape primitive. The most important difficulty, however, as we have already pointed out, is the complexity of the search. If no a priori knowledge about the shape, size or other attributes of such primitives is available, the space of all possible subgraphs [1] (of various sizes and configurations) of all the object graphs has to be searched for recurrence. Our algorithm avoids this complex search by breaking it into a hierarchical search. While the exhaustive search for all possible recurring subgraphs is intractable, our search in the space of whole objects first, and then in the space of object parts—and not all *random* subgraphs—proved to be feasible and effective, as shown below.

2.4.1. The learning algorithm

The learning algorithm consists of two stages. In the first stage, all objects are matched pairwise, and on the basis of the results, each object gets decomposed into its parts. This decomposition occurs strictly based on the activity history of nodes over the course of matching. Segmentation of an object part is a consequence of its nodes getting bound together due to their correlated pattern of match similarity over time (or across different matches). This segmentation method is very novel. It is based on higher-order statistics of local features (their matching pattern with other objects) rather than their direct first-order relationships (e.g., their similarity, or their fitting a pre-determined template such as corners, contours, surfaces, etc.), and it does not use any a priori knowledge about the patterns to be segmented. Even spatial contiguity is not assumed. Another advantage of this strategy is that only the recurring parts (i.e., shape primitives) get segmented, and not other parts of the objects. These segmented parts (subgraphs) are stored in memory as new object models. In the second stage, the same matching and binding process as in stage one operates on the new segmented parts. They all get matched

[1] We have previously shown (Shams, 1999) that already the number of contiguous subgraphs (which is a subset of the subgraphs we are interested in) of a planar graph (corresponding to our object graphs) is exponential in the number of graph nodes. Therefore, the number of subgraphs for *all* the object graphs will be prohibitively large.

with each other, and the parts of the same type (i.e., corresponding to the same shape primitive) get bound/fused together strictly based on their matching pattern. The structures that emerge from this process are what the algorithm has found as the common denominator in the composition of the objects in the memory. Below is a description of the algorithm in more detail.

2.4.1.1. Decomposition of composite objects into their parts. Each object is matched against all other objects. If the (highest) total similarity between objects k and l is below a pre-determined threshold t, the match is discarded. If it is above that threshold, the match is accepted, and M_i^{kl} will represent the similarity of jet i in object k (i.e., f_i^k) and its matching jet in object l (i.e., f_j^l).

$$M_i^{kl} = \tilde{s}(f_i^k, f_j^l); \quad \widetilde{S}(G_k, G_l) \geqslant t. \tag{5}$$

Success is achieved mostly when there is at least one shape primitive in common between the two objects and the match has found this correspondence. For successful matches see Fig. 4. Notice that this matching is not a trivial task, as a given shape primitive in general is partially occluded differently in different objects, lowering the total similarity. Also, even if the two matching shape primitives cause a very high similarity the match may still be discarded based on low total similarity $\widetilde{S}$ due to the presence of non-matching shape primitives or other parts. It turned out that despite getting some false positives due to the former problem, and discarding some good matches due to the latter, most matches were found correctly. Thus, sufficient statistics could be collected with a small number of objects (~25 objects for each primitive).

Now, to find, within the graph of an object k, regions that correspond to shape primitives, consider its jet match similarities M_i^{kl} as a function of the running index l; that is, the set of similarities of jet i for different matched objects l. We denote this vector by $\mathbf{M}_i^k$ (Fig. 5). It is to be expected that two jets i and j within a shape primitive region in object k either both have strong similarity with their corresponding matched jets (when the shape primitive region found a matching shape primitive in another object graph) or both have low similarity (when no such match was found). To detect such correlated patterns of similarity, pairwise correlations [2] are calculated as follows

$$R_{ij}^k = \frac{\mathbf{M}_i^{k\mathrm{T}} \mathbf{M}_j^k}{\left\|\mathbf{M}_i^k\right\| \cdot \left\|\mathbf{M}_j^k\right\|} \tag{6}$$

Fig. 6, bottom right, shows this correlation matrix obtained for the object shown on the top left. There are 54 jets in this object and we therefore have a 54×54 correlation matrix. It should be emphasized that the correlation value R_{ij}^k does not reflect whether two jets i and j of a given object graph are similar to each other or not; it rather reflects the correlation between the match similarity *history* of the two jets over the course of matching with other objects. We found that jets i and j of an object graph k—which themselves may be very different from each other—have high correlation R_{ij}^k if they are part of the same shape primitive, and have very low correlation R_{ij}^k if they fall on two different parts, even if they are very similar themselves. Therefore, by grouping and binding high match-history-correlation nodes together, the shape primitives in each object can be segmented out.

The binding of the jets together is achieved using a simple clustering algorithm, described in Appendix A. This algorithm takes the correlations R_{ij}^k as input, and outputs one or more clusters of graph nodes. The correlation matrix displayed in Fig. 6 yields, for example, two clusters, one consisting of nodes 1–38 (corresponding to the cylinder shape primitive) and one consisting of nodes 39–54 (corresponding to the cone shape primitive). This type of clustering which is based on pairwise (or relative) values as opposed to individual absolute quantities, is an NP-complete problem (see Appendix A for the reasoning). Our simple and stochastic clustering algorithm, however, by relaxing a restriction, succeeds in finding the optimal solution, and it does so very efficiently (in polynomial time).

Each segmented part is now saved as a new model graph.

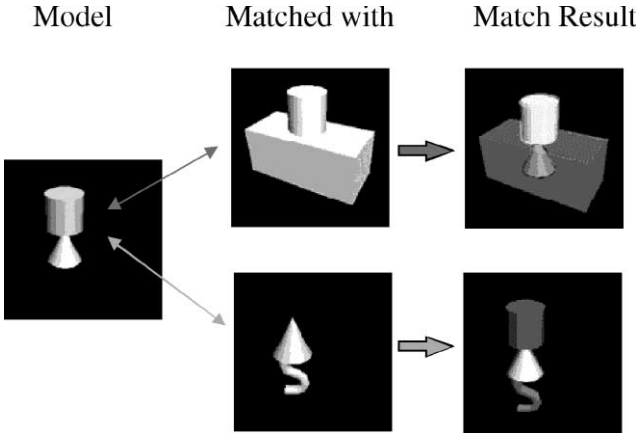

Fig. 4. Two schematic examples of successful matching. The model (left) is matched against objects (middle column). Matching results are shown in the right column. Bright areas represent the areas with high similarity.

[2] Notice that the correlation function used here is the same as that used for computing the similarity/correlation between jets (shown in Eq. (1)). We refer to it as "correlation" only in an intuitive sense. Technically, this formula does not correspond to the linear correlation, as the vectors are not mean-free.

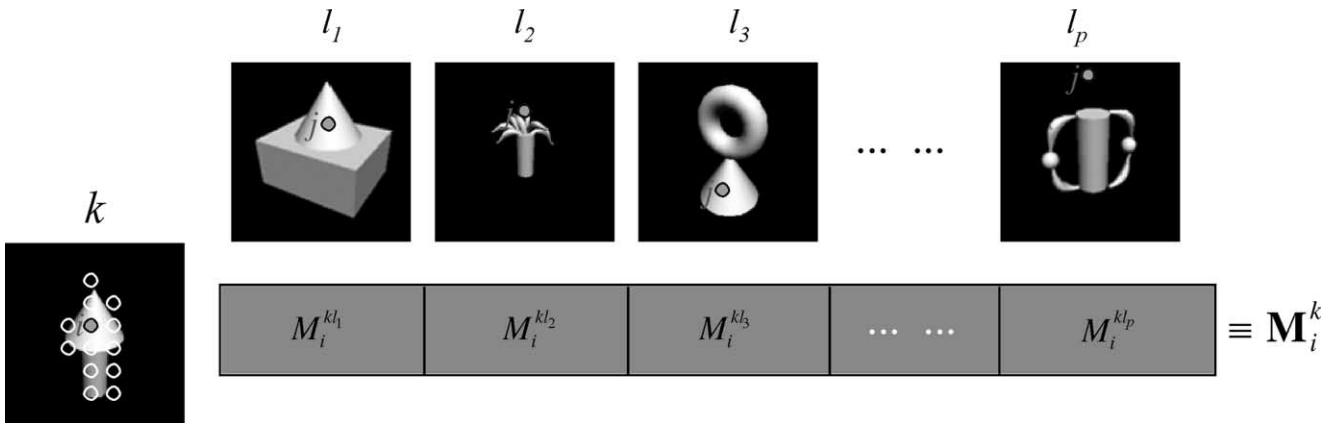

Fig. 5. History of jet similarities across matches. Vector $\mathbf{M}_i^k$ represents the similarity of the jet at node i with its corresponding jet (at varying coordinate j) in different matches.

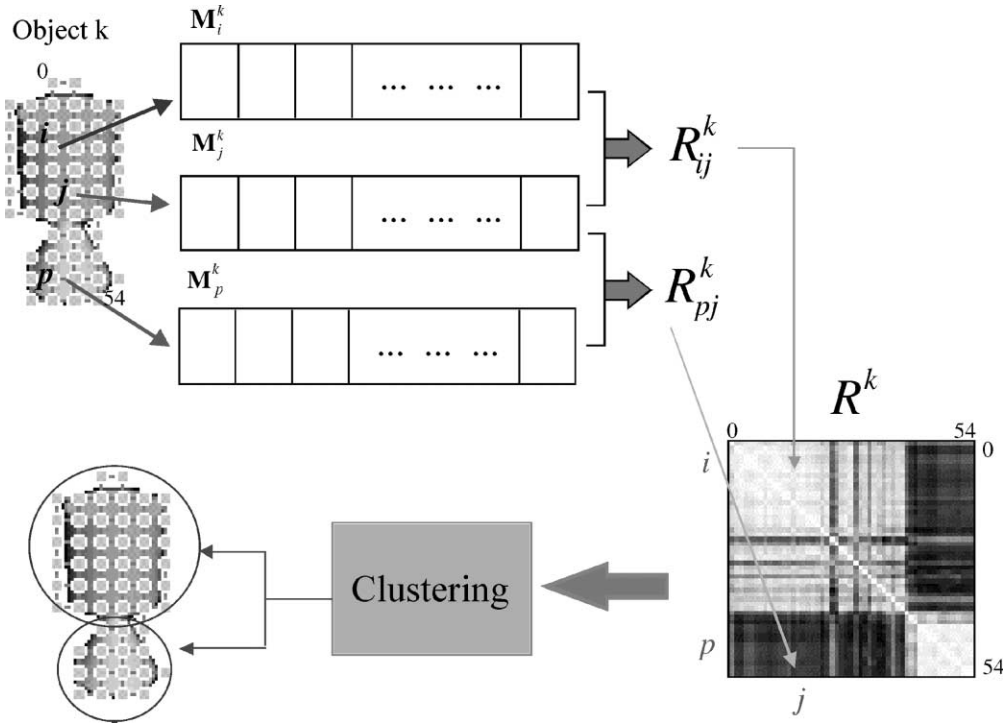

Fig. 6. An object composed of two shape primitives, with superimposed graph (left). The correlation matrix R^k for the same object graph is shown in the bottom right. Each axis of the matrix corresponds to the list of jets in the object graph. Two jets turn out to have high match history correlation if they belong to the same shape primitive (e.g., R_{ij}^k) and low match history correlation if they belong to different object parts—primitive or else—(e.g., R_{jp}^k).

2.4.1.2. Emergence of final learned patterns from segmented parts. Now that a new set of object graphs has been added to the database of our object models, stage I is essentially repeated, this time, however, using the new segmented graphs instead of the original whole-object graphs. All segmented parts are matched with each other and a match matrix is recorded:

$$m_{pq} = \widetilde{S}(g_p, g_q) \tag{7}$$

where g_p and g_q represent the segmented parts resulted from the decomposition of the all object graphs. Next, we compute the cross-correlation between the match similarities of pairs of segmented parts:

$$r_{pq} = \frac{\mathbf{m}_p^{\mathrm{T}}\mathbf{m}_q}{\|\mathbf{m}_p\|.\|\mathbf{m}_q\|} \tag{8}$$

where $\mathbf{m}_x$ denotes a vector of m_{xy} with running index y. Segmented parts which are inter-correlated in terms of their matching pattern get clustered together, when the clustering algorithm (cf. Appendix A) is applied to this matrix.

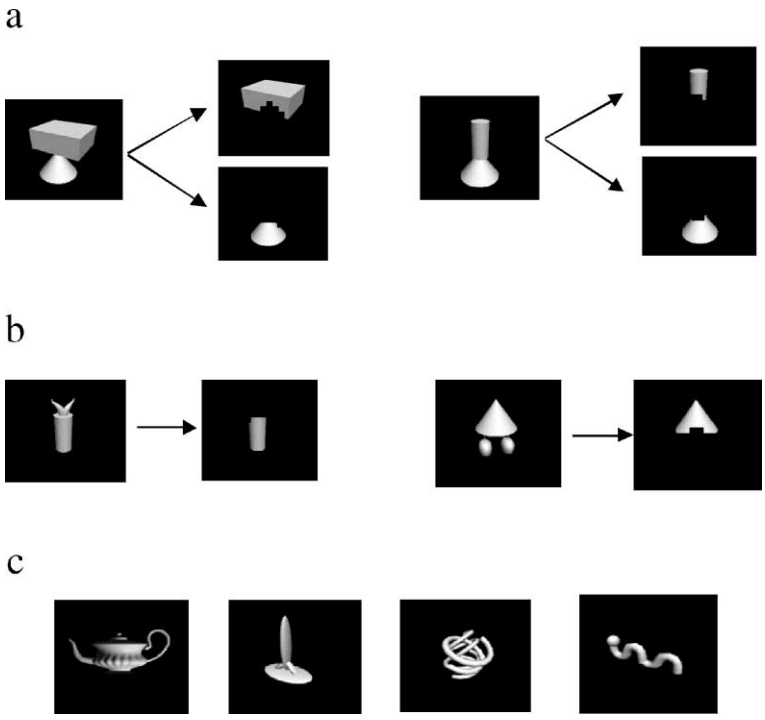

Fig. 7. Decomposition of objects into parts (or "partial shape primitives"): (a) decomposition of an object which is composed of two shape primitives results in two partial shape primitives, (b) decomposition of an object which is composed of one shape primitive and some other parts results in one partial shape primitive, and (c) decomposition of an object which is not composed of any shape primitives results in no partial shape primitives, i.e., no decomposition takes place.

The model graphs within one cluster are fused together in the relative position where they were matched, and by averaging the jets coming to lay on top of each other. For each cluster, the resulting graph composed of averaged jets is a complete representation of a shape primitive.

2.5. Results

2.5.1. Results of the shape primitive learning simulation

Applying the algorithm outlined above to the set of 104 objects, some of which are shown in Fig. 1, led to the results shown below. The decomposition of objects in the first stage of the algorithm resulted in segmented parts. As can be seen [3] in Fig. 7, these parts are in the form of shape primitives with missing portions due to the partial occlusions caused by other parts in the objects.

In the second stage, the segmented parts got clustered together, resulting in three clusters. Each cluster corresponded to a complete set of segmented parts of the same type (i.e., corresponding to the same shape primitive). Fig. 8 shows three examples of parts which belong to the same cluster.

Finally, Fig. 9 shows the three emerged patterns resulted from fusing the clustered parts together. The integration of parts within a cluster indeed formed a complete shape primitive, since each part of a given shape primitive is visible in at least one of the partial shape primitives. For each cluster, the resulting graph composed of averaged jets was a complete representation of a shape primitive. The structures that emerged from the learning process are indeed the three patterns that recurred in the composition of objects in the database: a cone, a cylinder, and a cube. All, and nothing but, the three recurring shape primitives were learned.

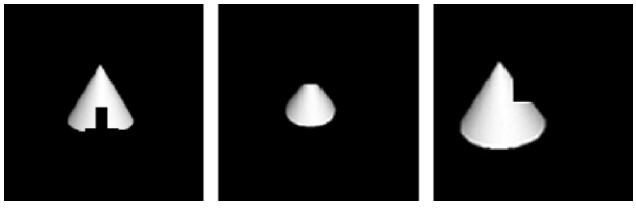

Fig. 8. Reconstructions of three partial shape primitives belonging to a cluster of parts. Each cluster contains as many parts as there are objects containing the corresponding primitive (~26).

[3] For diagnostic and display purposes we reconstructed images from graphs, by the following procedure. For each jet incorporated into a graph we went back to the point in the original image at which the jet had been extracted, and excised pixel values from a little square region around that point. When dealing with an aggregate graph, we then averaged the pixel values corresponding to the jets being averaged. Figs. 7–9, 13b and 15 represent such reconstructions. We used this method for reconstruction, since we have discarded phase information in the step of going from Gabor components to magnitudes thereof, and the reconstruction in the absence of phase information is computationally expensive (Shams & von der Malsburg, in press).

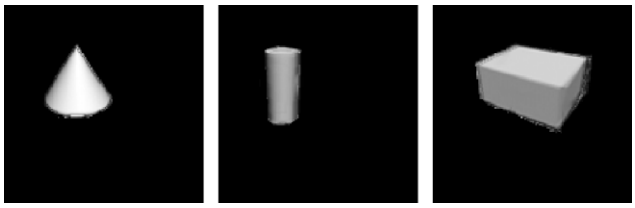

Fig. 9. Reconstructions of the three learned patterns by the algorithm. As can be seen, the patterns indeed correspond to the three shape primitives which took part in the composition of the objects in the database.

2.5.1.1. Analysis of the results. Each of the learned patterns is a lattice of averaged jets. The reconstruction of the three learned patterns, shown in Fig. 9, illustrates that the three learned patterns are indeed the three shape primitives. These images, however, contain imperfections (e.g., fuzziness around some contours, etc.) which we attribute to artifacts resulting from sparse spatial sampling. Each learned part image is naturally of a fixed size and geometric proportion, and therefore does not reveal any information as to the possible flexibility of the learned representations in terms of size variance. As the goal of learning shape primitive patterns is to use them to represent the objects which are composed of such shapes, it is important to test the effectiveness of the learned patterns in detection and discrimination of the shape primitives within objects. In this section, we present a functional examination of the learned patterns.

We emphasized earlier the distinction between the learning and the recognition of shape primitives. We stated that the goal of our model is not to simulate the detection or recognition of these shape primitives but rather the learning of them. In this section, we nevertheless focus on detection and recognition of shape primitives. We measure the effectiveness of the learned shape primitives in terms of detection and discrimination by comparing their performance with those of idealized shape primitives. By idealized shape primitives we refer to the prototype shapes (cube, cone, cylinder) which we used to generate the composite objects. The same prototype shape primitives were used in construction of all the objects which contained shape primitives, by combining them with other primitive or non-primitive parts, and varying scale. If the learned "shape primitives" are truly representative of shape primitives, then their recognition performance should resemble the recognition performance of the idealized shape primitives closely.

Fig. 10 shows these comparisons. The recognition performance of each learned shape primitive is plotted

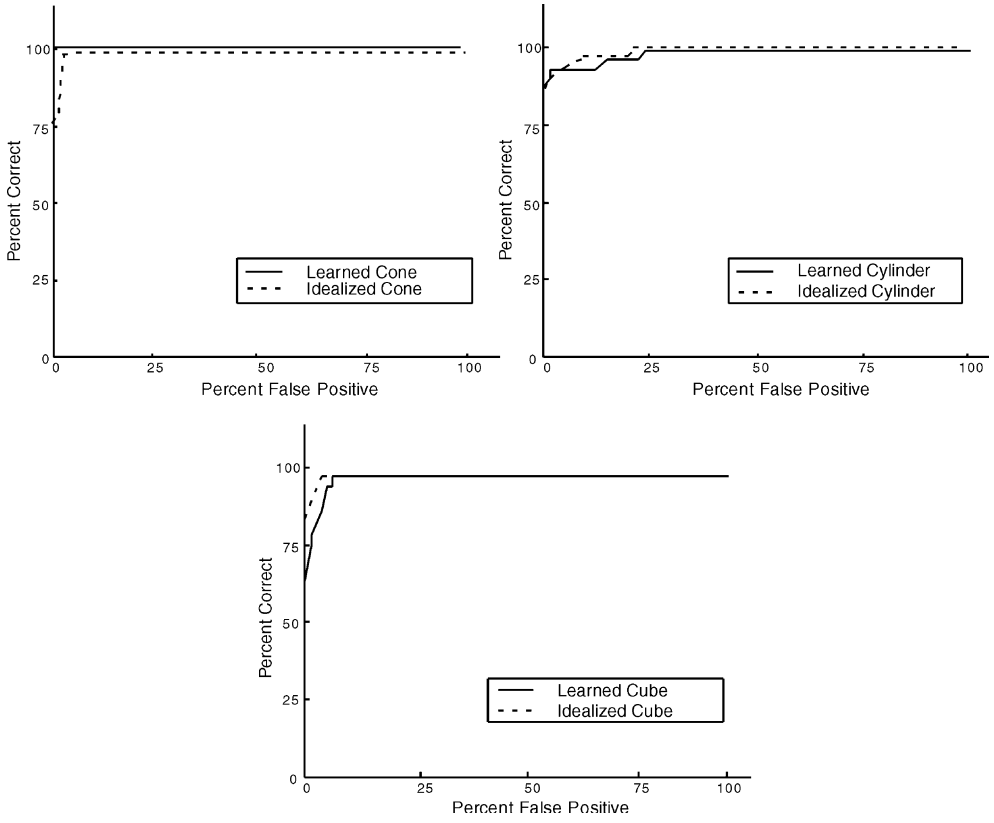

Fig. 10. Comparison of the recognition performance of the learned shape primitives with that of idealized shape primitives. The top left, top right, and bottom figures display the performance of cone, cylinder, and cube shape primitives, respectively. The solid lines represent the data from the learned shape primitives, and the broken lines represent those of idealized shape primitives.

against that of its corresponding idealized shape primitive.

Each graph displays the result of matching a learned shape primitive and an idealized shape primitive with 104 objects. In an ideal case, matching a shape primitive against the set of all objects would result in correct detection (i.e., finding the correct alignment) in the objects which contain the shape primitive, and would lead to total similarity values which are much lower than those of the correct matches, in the rest of the objects.

As it can be clearly seen in Fig. 10, the detection and discrimination power of the learned shape primitives is very high and very similar to that of the ideal case. Surprisingly, the learned cone outperforms the idealized cone. This is probably because the learned cone is composed of jets that have been averaged over several cones and hence allowing, on average, a better match to the varying examples of cones in the database, compared to the idealized cone's jets which are not averaged, and thus represent only one single cone.

3. Generality of the learning algorithm

Our learning algorithm does not explicitly take advantage of any specific features of the shape primitives. The essential property subserving learning is the recurrence of the shape primitive patterns in various objects, and this seems to be the only required attribute. There is, however, a possibility that the inherent configuration of the shape primitives, i.e., their composition of non-accidental features (e.g., vertices, parallel lines, smoothness, etc.) implicitly helps the learning process by rendering them more distinct from the non-shape primitive background (by background we refer to the various parts they are combined with in various objects), and giving them salient Gabor response signatures.

We investigated this question by testing our algorithm on a data set consisting of "irregular" objects and scenes. This was absolutely a post hoc test in that the stimuli used in this test were not used for development and testing of the algorithm before.

3.1. New stimuli

A good test data set should meet a few criteria. The patterns to be learned should be highly irregular and not shape primitive-like. On the other hand, to be relevant for the study of shape learning and recognition, their forms should not be unrealistic such as fractal or confetti patterns. The structures, called *digital embryos* (Brady, 1999), are generated by a stochastic process which is modeled after the embryological process. As it can be seen in the two examples shown in Fig. 11, they are highly irregular, and radically different from our other stimuli. In the meantime, they provide an appropriate test bed for the study of biological vision, as they resemble plants and animals.

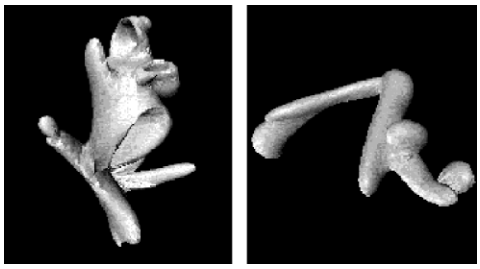

Fig. 11. Two examples of digital embryos. These objects are generated through a dynamic stochastic process modeled after the embryological process. Although the shapes are completely novel, they are not irrelevant to our visual recognition process as they resemble the shape of animals and plants.

In each image, one of the embryos to be learned is embedded in a background consisting of clutter of various other embryos. The irregular form of the embryos and the cluttered background consisting of the same type of shapes makes the "foreground" embryo indistinct and non-segmentable from the background based on any single scene. Although each embryo is a 3D synthetic object, approximately the same viewpoint is used in all the scenes. In different scenes, because of translation of the model within the scene, minor variations in orientation in depth are present due to the change in the relative position of the object to the camera. Some examples of the scenes are displayed in Fig. 12. It can be seen that the viewpoint and size of the embryo are fairly constant, however the position varies from one scene to another. The input to the algorithm is 25 such scenes all containing the embryo displayed in Fig. 13a. It should be noted that the images were generated by the inventor of digital embryos, in a completely automated and randomized fashion.

While there is technically no partial occlusion of the recurring embryo in any of the scenes, in reality, many contours of the object are missing, in varying locations in different scenes due to the blend with the background. This phenomenon acts similar to partial occlusion, and is somewhat comparable to the varying partial occlusions which existed in the shape primitive dataset. This dataset lacks the size variation which existed in the shape primitive dataset, but on the other hand, contains two other sources of difficulty. Firstly, the scenes contain backgrounds which are much more complex than those in the shape primitive database, and thus can produce significant noise in the Gabor-wavelet responses of the objects due to background interference. Also, the recurring pattern is not monolithic, as opposed to shape primitives, and is composed of several parts. Some of these parts occur in the background in some scenes, and therefore, can potentially misguide the matching process.

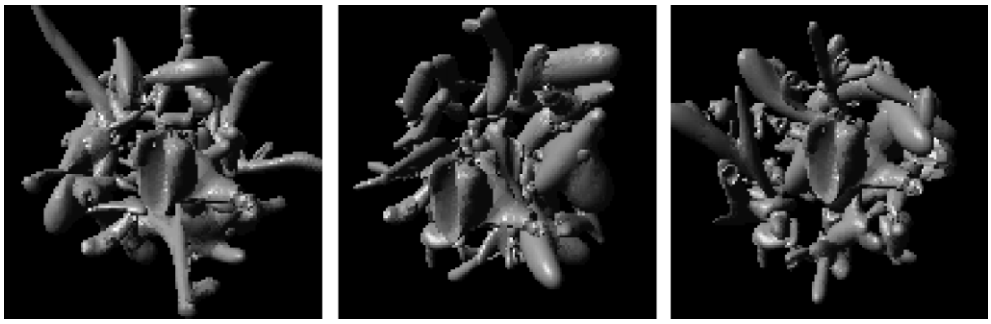

Fig. 12. Examples of scenes which served as input to the model. In each scene the image of recurring embryo appears on a background cluttered with numerous other embryos. While the foreground embryo (i.e., the recurring embryo is not distinguishable from the background based on a single scene), it becomes distinct and segmentable in our eyes after exposure to a number of such scenes. The question to be examined here is whether the learning algorithm is also able to extract the recurring embryo based merely on a set of such scenes.

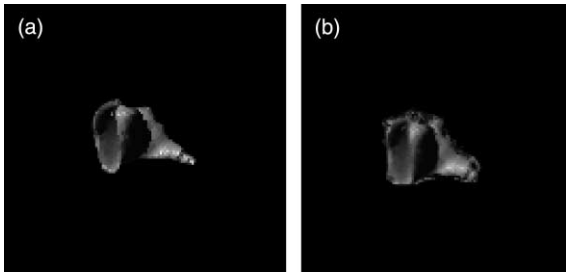

Fig. 13. (a) The recurring embryo. This image was embedded in various backgrounds composed of numerous other embryos in each scene, in varying positions within the scene and (b) the reconstruction of the learned pattern.

As can be seen in the pictures of Fig. 12, the embryo scene dataset is very different from the shape primitive dataset (Fig. 1) used before. Such radical change in set of stimuli usually requires tuning of all the parameters and even modifications in the algorithm to adapt to the new stimuli. Surprisingly, however, we found that nothing needed to be changed in our algorithm. The parameters and procedures of the model remained the same in our embryo learning simulations, and the results presented below are the outcome of the algorithm as described above, without any customization for embryo dataset.

3.2. Results

The learning algorithm took as input 25 scenes (examples shown in Fig. 12) containing the recurring embryo shown in Fig. 13a, and outputted the pattern illustrated in Fig. 13b. As can be seen, the learned pattern represents the recurring embryo quite well.

We have tested the algorithm on two other embryos each embedded in 25 cluttered embryo scenes, and they were also successfully learned. The three embryos we tested the algorithm with were not hand picked, and thus, the results can be taken as typical results. Considering that the new set of stimuli were radically different from the original stimuli, that this test was purely post-hoc, that the new stimuli were not hand-picked, and that no changes were needed neither in the algorithm nor in the parameter set, the success of the algorithm in these simulations indicates powerful generality of the algorithm.

4. Discussion

4.1. Relationship to other models

In this section, we compare and contrast our model with previous successful unsupervised models that learn features from complex and realistic images (Amit & Geman, 1999; Bell & Sejnowski, 1997; Krüger, 1998; Lee & Seung, 1999; Olshausen & Field, 1996; Weber, Welling, & Perona, 2000). These previous models can be divided into two categories: object-model learning algorithms and feature learning algorithms. These two categories address two different questions. The object-model learning algorithms learn features that are intended for building a *complete model* for *one specific object class*, whereas the feature learning algorithms learn *parts* that occur *across various object classes*. Parts occur in *some* of the objects in the training set in varying positions, and once learned, serve as intermediate representation that can be used in various topological relationships with other parts to represent novel or familiar objects. In contrast, the features learned in the object-model-learning algorithms occur always in the same topological relationship with other features within the object. For example, while a face-model learning algorithm may learn several different types of mouths, each mouth will always be combined with the same number of other features (e.g., two eyes and one nose) and in the same topological relationship to represent a novel or familiar face.

Our algorithm falls in the category of feature learning algorithms, and in this way, it is distinct from

model-learning algorithms in a fundamental way. Examples of object-model-learning algorithms that are the most related to our work are the algorithms of Weber et al. (2000), Amit and Geman (1999), and Lee and Seung (1999). Weber et al.'s algorithm learns models for faces or cars. Lee and Seung's algorithm learns a model for faces. Amit and Geman's algorithm learns a model for faces or letter Zs. In all of these models, the features that are learned are extracted from images within one object class, and it is not clear whether they will be useful as intermediate representation for other object classes. Besides the fundamental difference of belonging to two different categories, there are also other significant differences between our model and these models. For example, input images used by Weber et al. are more complex than ours, but their algorithm contains a supervised component: while the target objects are not segmented or marked within the image, and the candidate features are automatically extracted in an unsupervised fashion, the algorithm, however, uses a validation set which is composed of labeled images (target vs. no-target) in order to select the best subset of features. Furthermore, the number and the size of features to be learned are set a priori and the algorithm is only invariant to translation and not to other affine transformations. Similarly, in Lee and Seung's model, the number of features is fixed a priori and the algorithm does not offer any affine transform invariance (including translation). Besides these differences, however, some similarities are noteworthy between our model and those of Weber et al. and Amit and Geman. Similar to their algorithms, our model also heavily exploits topological information. The object models learned by Weber et al. and Amit and Geman are highly sparse, similar to the graph representation used in our algorithm. The "local move", implemented in our graph matching process to allow for local distortions, is related to Amit and Geman's circular subregions in the edge-groupings or global arrangements.

Our algorithm is in essence much closer to the other 'feature learning' algorithms (Bell & Sejnowski, 1997; Krüger, 1998; Olshausen & Field, 1996) where features are learned from a training set consisting of various object classes, and thus, the learned features can take part in partial representation of any arbitrary object which may happen to contain that feature in any arbitrary topological location. These models are discussed briefly in the next section.

4.2. Contributions of the model

Our model is the first model that offers a successful scheme for unsupervised learning of shape primitives, and it does so without using any a priori assumptions about the structure, number, size, location, or any other attribute of these patterns.

The success of the previous unsupervised shape feature learning algorithms remains largely within the realm of very low level features such as oriented edge/bar detectors (Bell & Sejnowski, 1997; Olshausen & Field, 1996), curve collinearity and parallelism (Krüger, 1998). Many researchers have sought to learn vertices from images but none has succeeded. Perhaps the reason for the lack of notable success in learning more complex features, or higher level representations, is partly due to the fact that derivation of such features involves extraction of *higher* order—in contrast to second order—correlations among the lower level features. Our algorithm extracts these higher order correlations among the features (jets) by simply interleaving two linear correlation methods; that is, by finding the correlation among the jet similarities [4] which are in themselves the measures of correlation between jets.

The two main operations underlying our learning algorithm—graph matching and (Hebbian type) linear cross-correlation—as well as the features and object representations used, are biologically plausible. [5] Furthermore, the model lends itself to a parallel distributed implementation. Each object model is matched with other models, gets decomposed, and each partial shape primitive model gets matched with other partial shape primitive models. All these processes can be naturally performed for all object (or partial shape primitive) models in parallel.

Learning of shape primitives from composite objects—as are most objects in the world—requires generalization over examples which vary in a number of aspects. Some of these possible variations are in partial occlusion, position, size, and aspect ratio. Our learning model copes with all these variations. Slight variation existed also in the orientation in depth of the shape primitives across various objects. The lighting, too, varied to some extent, as it was not controlled from one image to another. Despite the difficulties introduced by such variations and despite the complexity of the learning problem, our algorithm performed in reasonable time (polynomial time complexity), and a parallel implementation of the algorithm, as exists in the brain, would make the processing time even shorter.

In most models the number of the required training samples grows exponentially with the number of dimensions (e.g., size, position, etc.) over which the model

[4] By this we refer to the correlation between jet similarity histories $\mathbf{M}_i^k$.

[5] Biological plausibility of graph matching and lateral excitation are discussed in detail in Shams (1999). Long-term potentiation and long-term depression are examples of cross-correlation operation in the brain (Bear & Malenka, 1994; Bliss & Lomo, 1973). Biological plausibility of the features and object representations was discussed in Section 2.2.

has to generalize. Another well-known problem in neural networks is the "curse of dimensionality" or the exponential increase in the needed sample size as a function of input dimensions. This is why most neural network models utilize toy-like stimuli as input (e.g., an 8×8 matrix of binary pixels). Our input, on the other hand, consists of images typically in the order of 100×100 128-gray-level pixels, and the real-valued Gabor transform increases this dimensionality substantially further. Due to the high information content of our input (composite objects), even the most clever preprocessing (e.g., an ideal edge representation) could not reduce this dimensionality sufficiently to avoid the requirement of an astronomically large training sample by ordinary methods. In our model, despite the several dimensions of variability (e.g., size, occlusion, position, etc.), as well as the large information content of the input, the number of needed examples was surprisingly low: a total of 104 objects, one fourth of which did not contain any shape primitives. We did not experiment with smaller numbers of objects, and it is possible that the needed number of examples is even lower.

Most notably, our model rests on a very simple premise, which as a result expands the application domain of it. As pointed out earlier, the model does not utilize any constraints about the configuration or structure of neither the input objects nor the patterns to be learned. More specifically, no structural feature of the shape primitives is explicitly exploited by the model. The only apparent distinction between the shape primitives and other object parts was the *recurrence* of these structures. If the only premise in the model is that any recurring pattern (regardless of its specific configuration) can emerge as a new complex feature/representation, then the model should be applicable to any type of recurring patterns and not just to regular parts such as shape primitives. This was indeed verified through the simulations presented in the previous section where irregular, complex shaped patterns (digital embryos) were learned based on a small number of complex scenes in which they recurred.

It should be pointed out that while our learning model makes no a priori assumptions about any attributes of the individual patterns to be learned, it does make an inherent assumption about the *equality* of pairs of patterns: two patterns are equal if they are matchable by our elastic matching mechanism which allows the aforementioned variations. The elastic matching process provides a great advantage for our approach. While the traditional bottom-up approaches would have to learn (through combinatorial number of training examples) the aforementioned invariances in the input space, our model is equipped with a matching mechanism that readily provides these invariances from the beginning.

Below we describe a new principle which we believe endows our model with its learning capability.

A single repeat of a subpattern can make it stand it out as a significant feature, leading to a very efficient learning from very few examples. Although this type of learning is a common trait of the brain, artificial systems have been unable to replicate it, always requiring a very large number of training examples for learning any kind of features. We suggest that in order to learn new visual features from few examples, topological information embedded in the patterns within the scenes has to be exploited in a more effective way. Topological information offers a very rich source of information and can salvage the problem of finding matching patterns in an astronomical space of subpatterns and in presence of variations they undergo. To exploit this information, the sought subpatterns must be appropriately large (to make them "non-accidental"), and the matching process must be sufficiently flexible and powerful to cope with variations. This strategy can make the extraction of significant recurring patterns in the scenes computationally feasible.

Our algorithm uses this principle twice in succession. The segmented parts that result from the first stage are those subgraphs (out of a million other subgraphs) that stood out as significant because of a few repeats across objects. In the second stage of the algorithm, the segmented parts get clustered together and emerge as the final pattern due to a few repeats of each of the shape primitives across the segmented parts.

4.3. Unaddressed issues and future directions

We have been referring to 3D structures such as cones and cylinders as shape primitives, nonetheless, presented a model for learning of 2D patterns. In effect, the work presented in this paper has focused on 2D projections of the 3D shape primitives. As the problem of learning intermediate representations such as shape primitives has not been directly tackled up until now, a natural first step in this direction was to attack the problem of learning the 2D patterns first. It is not unlikely that the extension of the learning model presented here to 3D learning would only involve using 3D object representations as input and performing graph matching on 3D representations as opposed to single views. Such a 3D representation could simply consist of a collection of various projections of each object, together with a mapping between corresponding points across the various views. Matching a pair of object models in this scheme would, in the worst case, involve matching all views of the two objects with each other. An effective and biologically plausible 3D representation of objects is still an open research issue.

It would be very desirable for a learning model to generalize over the parts which are affine transformations of each other. Affine transforms involve four types of operations: translation, scaling, rotation, and

shearing. Our results clearly show that the algorithm is able to cope with translation (in plane) and scaling. The various shape primitive examples in our database are of the same orientation in plane. In the real world, objects may be rotated, and a learning system should ideally be able to cope with this type of variation. This problem is in essence very similar to the problem of size variation. Invariance to orientation in plane can be achieved by adding another degree of freedom to the graph matching procedure, such that a graph can be rotated in addition to translation and rescaling. We have shown this type of invariance, i.e., invariance through flexible matching, to work for size variations. We have used this same method also to cope with shearing. We incorporated in the graph matching two size dimensions, horizontal and vertical, which could vary independently of each other. This mechanism allows variation in aspect ratio, e.g., allowing a match between a short squat cone with a tall thin cone. We have already shown that the model is able to find the correct aspect ratio, within the space of all aspect ratios that are examined. Although the search space of matching was increased to allow this variation, and the algorithm was shown to be able to find the correct aspect ratio in the matching process, this ability is not explicitly illustrated in our results as the database did not contain shape primitives of varying aspect ratios. In other words, although the algorithm is able to cope with shearing, we did not explicitly take advantage of this robustness in this particular database of objects.

The object images used in our model are segmented and lack texture or surface markings. In other words, our algorithm operates on input which is similar to a depth profile in information content. It is likely that this information can be provided by a combination of cues such as motion (object or viewer), stereo, color and the edge information (see Shams (1999) for a discussion), cues which are typically available in our natural encounter with objects. We have previously proposed an architecture that may be capable of performing this task (Shams, 1999), however, this architecture has not been implemented yet.

Finally, apart from partial occlusions, the examples of shape primitives in our database all have idealized shapes. There are no significant irregularities in any of them. In the real world, the shape primitives constituting objects are not always completely regular and prototypical. They may contain various irregularities in various points. This can potentially make the task of finding the correct correspondence between two shape primitives in two objects difficult. The local move, which allows the optimization of the match in individual nodes (see Section 2.4.1) during matching, provides robustness to local distortions. We already used this feature in our model to cope with slight imperfections and inconsistencies between two examples of a given shape primitive. It is to be tested whether or not this feature of the matching process is capable of coping with higher degrees of distortion.

Our model makes several predictions that may be tested experimentally: (a) As the model relies on topological information in extracting recurring patterns, it is predicted that learning large patterns is easier than learning small ones. Because finding partial matches is less reliable for small patterns, it is expected that finding the correct match will be harder for smaller patterns, may lead to more failures, and thus a larger number of examples will be needed for learning to occur. (b) Similarly, we expect that learning contiguous patterns to be easier than patterns that are composed of spatially detached parts or contain concavities. For example, learning a thin doughnut is expected to be harder than learning a disk. This prediction is based on the role of lateral excitation in the process of graph matching. Lateral excitation is more effective for recurring patterns that are contiguous. (c) As mentioned above, the algorithm's robustness to distortion is provided by 'local move' of the individual nodes during graph matching. While this mechanism allows some distortion invariance (as evident by the results), it will clearly fail if the degree of local distortions is consistently larger than the distance between two neighboring jets in the graph (i.e., disturbing the topological relationships). Therefore, our model would predict that distorted images that would cause our algorithm to fail would not be learnable by human subjects either. (d) An extended version of our algorithm, which would be able to cope with variation in 3D viewpoint, would generally require at least several views of each object (if composed of non-symmetric shape primitives) in order for the matching process to find the partial matches between the corresponding views of the shared primitives. Thus, it is predicted that given only a single snapshot of each object, the novel recurring shapes (or shape primitives) would not be learnable; and at least multiple views of each object would be typically necessary for the acquisition of the recurring 3D patterns. (e) Finally, as the basis for learning is assumed to be matching of all stored patterns (objects or scenes etc.) with each other, we expect great variation in learning performance across individuals in learning a given pattern, as their life-long experience and hence their database of stored patterns are different. Some individuals may need five training samples while others may require 20 or more.

Acknowledgements

We thank Stefan Schaal and Yukiyasu Kamitani for their insightful comments on the manuscript. This work was supported by NSF's IMSC program at USC, and NIH grant HD08506.

Appendix A. Decomposition of an object into parts by feature binding

In this section we discuss the problem of *binding* together a group of entities, given pairwise relations (e.g., in the form of correlations) between them. Another interpretation of this problem is to extract higher-order correlations from second-order ones. This task seems to play an important role in many brain functions (from perception to cognition), and is a non-trivial problem.

This problem is in essence a clustering problem, and in the clustering literature is known as *pairwise data clustering* (Buhmann & Hofmann, 1994; Hofmann & Buhmann, 1995, 1997) or *proximity based clustering* (Puzicha, Hofmann, & Buhmann, 2000). It refers to clustering of entities based on *similarity/dissimilarity information* between *pairs* of entities, as opposed to absolute information on individual entities.

Alternatively, the problem of binding the object nodes together based on their pairwise data can be framed in a graph context: the features to be bound together interpreted as the graph nodes, and the pairwise data interpreted as the graph links. The problem of finding intercorrelated groups of features would be equivalent to the problem finding cliques in the graph. A clique is the largest fully interconnected subgraph in a graph. The clique finding problem is an NP-complete problem (Cormen, Leiserson, & Rivest, 1990), but below we describe an optimization algorithm which finds a solution to a less strict variation of the problem in polynomial time.

We used clique finding to bind the jets belonging to the same object part (shape primitive) together based on the pairwise cross-correlation between their similarity values across various matches. By finding cliques of jets within an object graph we decompose the object into its parts, or more precisely, we extract the shape primitives within an object and discard the non-primitive parts which would correspond to no cliques. If an object is made up of two shape primitives (and perhaps some other parts), it should result in two cliques; if it is composed of one shape primitive (and some other parts) it should lead to only one clique, and if it does not contain any shape primitives it should result in *no* cliques. Noise makes the problem of finding cliques potentially more difficult by introducing spurious edges between the cluster members and the nodes outside the cluster, as well as missing edges between some nodes within the cluster.

We describe an algorithm for solving the clique problem outlined above which is in a dynamical systems framework. The intuition behind this heuristic is as follows: interpret each node in the graph as a mass (uniform across different nodes), and imagine a gravitational force between those masses (nodes) which are connected to each other via an edge and no such force otherwise. (The gravitational force can be weighted by the inverse of the distance between the two masses. We found, however, that for our application this weighting is not necessary.) Such gravitational system should lead to the collapse of the nodes belonging to a clique into one point in space, due to the high connectivity among these nodes. On the other hand, the nodes which do not belong to any cliques would most likely get pulled by different sporadic masses to which they are connected but never strongly enough in one coherent direction so that they get absorbed by a clique. Thus, one can expect that the members of each clique in the graph would collapse into one point and the dynamics would converge to a state where the only remaining masses are those each consisting of a clique and those each consisting of one (or perhaps a couple) of the nodes which do not belong to any cliques. The cliques in this scheme can then be identified by excluding the masses which are composed of no more than a couple of nodes (cliques consisting of <3 nodes are not interesting). What remains is those masses which are the result of the collapse of several nodes into one point, each corresponding to a clique.

At each iteration, a node i is selected at random, and its position x_i is changed by

$$\Delta x_i = \alpha \frac{f_i}{|f_i|} \tag{A.1}$$

$$f_i = \sum_j \frac{d_{ij}}{|d_{ij}|} \tag{A.2}$$

where j represents the nodes which are connected to node i, and d_{ij} the vectors connecting node i to nodes j. Eq. (A.2) illustrates the net "force" which is exerted to node i by all nodes j. Each node j pulls node i towards itself (i.e., in the direction d_{ij}). The sum of such forces can add up to a large amount, pulling node i in the correct direction, however, far too much, moving to a position far beyond the location of nodes j. To prevent this, the amount of the move should be re-adjusted such that node i would not pass through the cloud of nodes j, but rather only get closer to them. To this end, we move node i in the direction of f_i however for a distance α which is half of the median distance between node i and nodes j (see Eq. (A.1)).

The choice of the criterion for stopping the iteration is not trivial. It turned out, however, that a very simple condition was highly adequate for our shape primitive decomposition application. The iteration was stopped as soon as a move (or a change in position) Δx_i became smaller than a threshold—a very small positive number. The algorithm is highly robust to this threshold. The reason for success of this convergence criterion is that when a move is nearly zero, all members of a given clique have already converged to the same position. This is the case because each member of the clique always

moves towards approximately the center of mass of the clique and the amount of the move is proportional to its distance to the clique members; the farther a node is from the center of the nodes *j* cloud, the larger the move. This way all clique members collapse into one point roughly at the same time. Thus, the stability of one node would signal the stability of all.

The objects in our database each contain zero, one, or two shape primitives, in addition to zero, one, or more non-primitive parts. Fig. 14 shows an example of an object which is composed of two shape primitives and no other parts, a cube and a cone. The top right image displays the position of the nodes at the beginning, before the iteration starts. The image on the left in the middle row shows the position of the nodes after the algorithm has converged. As can be seen, the nodes—which initially were spread out on a regular lattice—have now converged into two positions. The nodes which have converged to the top position are those corresponding to the jets falling on the cube, as displayed on the middle right image; and those converged to the lower point correspond to the jets falling on the cone, as seen in the bottom right image.

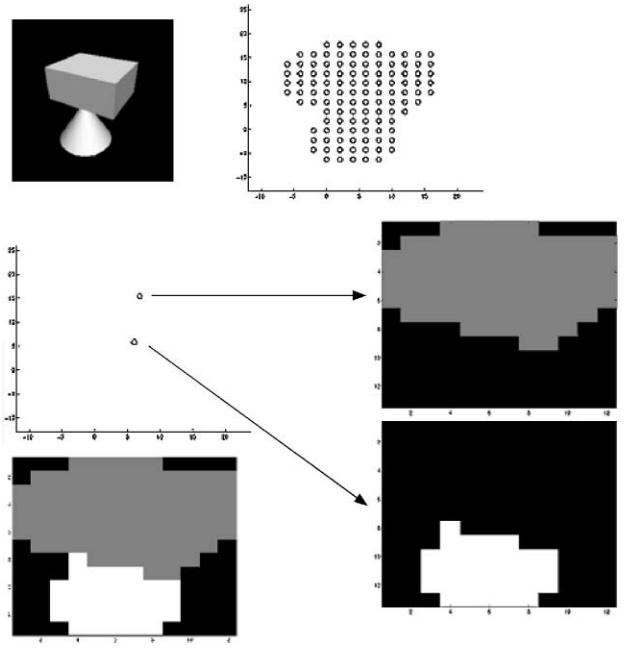

Fig. 14. Detection of *two* cliques within a graph using our algorithm. The image of an object consisting of two shape primitives (a cube and a cone), and its corresponding graph are displayed on the top row. To avoid clutter in the display, the edges in the graph which represent the pairwise correlations between the nodes are not displayed. The image on the left in the middle row, illustrates the final state of the graph nodes after the algorithm converges. The nodes in the graph have collapsed into only two positions, each representing a clique. Each clique corresponds to an object part displayed to the right. The image below the final state of the graph nodes shows the topological relationship between the clique members. The nodes belonging to the top clique are colored gray while the nodes belonging to the bottom clique are colored white.

If a jet is not sufficiently connected to the members of any of the cliques, it will remain standalone, and will not be absorbed by any of the cliques. This situation can be seen in the next example displayed in Fig. 15. The object in this example (displayed in the top left image) is composed of one shape primitive only—a cone. The top right image displays the position of the nodes in the initial state, and the bottom left image shows the node positions in the final state. This time only the nodes which correspond to the cone have collapsed into one point (the solid circle). The other nodes which correspond to the inverted S shape (a non-primitive part) have not clustered, and remain in their original position. The image on the bottom right is the reconstruction of the nodes which are members of the clique found. We measured the time complexity of the algorithm empirically across all the decomposed objects in the database, and found it to be polynomial in the number of nodes in the object graph.

Investigating the key element underlying the success of this algorithm, we examined the importance of the topological constraint, which is implicitly embedded in the algorithm in the initial state: the initial topographic position of the nodes are those of the object graph, as it can be seen in Figs. 14 and 15. To examine the role of topological information, we randomized the initial node positions. We found that, despite of this change, the algorithm converges to the same states as before. That

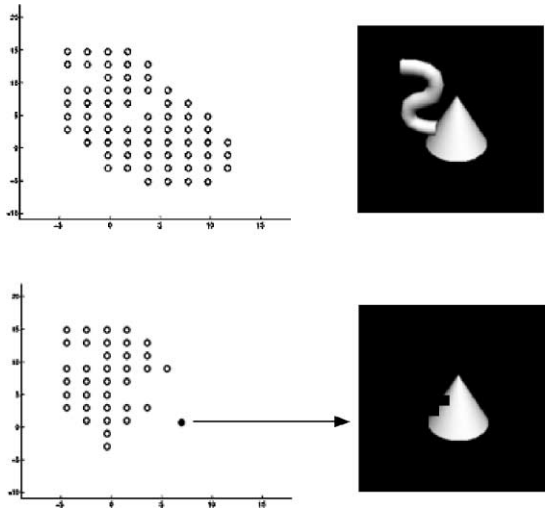

Fig. 15. Detection of *one* clique within a graph using our algorithm. The image of an object consisting of one shape primitive (a cone) is displayed on the right in the top row. To its left the node positions in the initial state of the algorithm are displayed and below that graph the final state of the position of the graph nodes after the algorithm has converged is displayed. The position which contains the clique members is the solid circle and is the only position containing more than one node. The clique (collapsed at this position) contains the nodes whose reconstruction is displayed in the image to the right.

is, the usage of correct topological positions in the initial state is not necessary for the algorithm to work. This finding is very interesting since it indicates that—at least for our application—the topological information seems to play no relevant role. If it is not the topological constraint that breaks the NP-completeness of this problem and makes it solvable in polynomial time, then what is?

We believe the answer is *noise tolerance*. As mentioned before, the jets falling within each shape primitive are highly intercorrelated with each other. However, there are usually some links (correlations above the threshold) which are missing within each shape primitive and there are also some spurious links between the jets falling on two different object parts (be it a shape primitive or not). These missing or spurious links amount to noise, and to be able to find the cliques correctly, the algorithm should be robust to such noise. While the existence of noise made the problem seem more complicated at the beginning, the results suggest the contrary. In our algorithm, for a node to cluster with a group of nodes it does not necessarily need to be connected to all of them. The clique problem, on the other hand, in the strict sense, requires a node to be connected to *all* of the nodes in a group for it to qualify as a member of the clique. Even if a node is connected to 99 nodes of a 100-node clique, it will not qualify to be added to the clique. It seems that it is the elimination of this strictness that makes the problem solvable in polynomial time.

References

Amit, Y., & Geman, D. (1999). A computational model for visual selection. *Neural Computation, 11*(7), 1691–1715.

Bar, M., & Biederman, I. (1995). *One-shot viewpoint invariance in matching novel objects*. Paper presented at the ARVO.

Barr, A. H. (1981). Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications, 1*(1), 11–23.

Bear, M. F., & Malenka, R. C. (1994). Synaptic plasticity: LTP and LTD. *Current Opinions in Neurobiology, 4*, 389–399.

Bell, A. J., & Sejnowski, T. J. (1997). The 'independent components' of natural scenes are edge filters. *Vision Research, 37*(23), 3327–3338.

Bergevin, R., & Levine, M. D. (1988). *Recognition of 3-D objects in 2-D line drawings: an approach based on geons*, Tech. Rep. TR-CIM-88-24, McGill University.

Biederman, I. (1987). Recognition-by-components: a theory of human understanding. *Psychological Review, 94*(2), 115–147.

Biederman, I., & Cooper, E. E. (1991a). Priming contour-deleted images: evidence for intermediate representations in visual object recognition. *Cognitive Psychology, 23*, 393–419.

Biederman, I., & Cooper, E. E. (1991b). Evidence for complete translational and reflectional invariance in visual object priming. *Perception, 20*, 585–593.

Biederman, I., & Gerhardstein, P. C. (1993). Recognizing depth-rotated objects: evidence and conditions for three-dimensional viewpoint invariance. *Journal of Experimental Psychology: Human Perception and Performance, 19*, 1162–1182.

Bienenstock, E., & von der Malsburg, C. (1987). A neural network for invariant pattern recognition. *Europhysics Letters, 4*, 121–126.

Bliss, P. V. P., & Lomo, T. (1973). Long-lasting potentiation of synaptic transmission in the dentate gyrus of the anesthetized rabbit following stimulation of the perforant path. *Journal of Physiology (London), 232*, 331–356.

Boult, T. E., & Gross, A. D. (1987). Recovery of superquadrics from 3D information. *SPIE Intelligent Robots and Computer Vision: Sixth in a Series, 848*.

Brady, M. (1999). *Psychophysical investigations of incomplete forms and forms with background*. Unpublished Ph.D. Thesis, University of Minnesota.

Brady, M. J. (1998). *Learning to recognize camouflaged novel objects*. Paper presented at the The Association for Research in Vision and Ophthalmology Meeting, Florida.

Brooks, R. A. (1983). Model-based three-dimensional interpretations of two-dimensional images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5*(2), 140–150.

Buhmann, J., & Hofmann, T. (1994). *A maximum entropy approach to pairwise data clustering*. Paper presented at the International Conference on Pattern Recognition, Hebrew University, Jerusalem.

Cooper, E. E. (1993). *Does global shape play a role in visual object priming?* University of Minnesota.

Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (1990). *Introduction to algorithms*. Cambridge: MIT Press.

Dickinson, S. J., Pentland, A. P., & Rosenfeld, A. (1992). 3-D shape recovery using distributed aspect matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 14*(2), 169–193.

Ferrie, F. P., Lagarde, J., & White, P. (1993). Darboux frames, snakes, and superquadrics. *IEEE Transactions Pattern Analysis and Machine Intelligence, 15*(8), 771–784.

Gilbert, C. D. (1992). Horizontal integration and cortical dynamics. *Neuron, 9*(1–13), 121–128.

Helmholtz, H. v. (1962). *Treatise on physiological optics*. New York: Dover.

Hofmann, T., & Buhmann, J. (1995). *Hierarchical pairwise data clustering by mean-field annealing*. Paper presented at the International Conference on Artificial Neural Networks.

Hofmann, T., & Buhmann, J. (1997). Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(1), 1–14.

Krüger, N. (1998). Collinearity and parallelism are statistically significant second order relations of complex cell responses. *Neural Processing Letters, 8*(2), 117–129.

Kumar, S., Han, S., Goldgof, D., & Bowyer, K. (1995). On recovering hyperquadrics from range data. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 17*(11), 1079–1083.

Lades, M. (1994). *Invariant object recognition with dynamical links, robust to variations in illumination*. Unpublished Ph.D. Thesis, Ruhr-Universität Bochum.

Lades, M., Vorbrüggen, J. C., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R. P., & Konen, W. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transaction on Computers, 42*(3), 300–311.

Lee, D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature, 401*(6755), 788–791.

Maurer, T., & von der Malsburg, C. (1995). *Learning feature transformations to recognize faces rotated in depth*. Paper presented at the International Conference on Artificial Neural Networks, Paris, France, 9–13 October.

Nevatia, R., & Binford, T. O. (1977). Description and recognition of complex curved objects. *Artificial Intelligence, 8*, 77–98.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive fields properties by learning a sparse code for natural images. *Nature, 381*(13), 607–609.

Puzicha, J., Hofmann, T., & Buhmann, J. (2000). A theory of proximity based clustering: structure detection by optimization. *Pattern Recognition, 33*(4), 617–663.

Raja, N. S., & Jain, A. K. (1992). Recognizing geons from superquadrics fitted to range data. *Imaging and Vision Computing*, 179–190.

Schyns, P. G., & Murphy, G. L. (1994). The ontogeny of part representation in object concepts. In D. L. Medin (Ed.), *The psychology of learning and motivation* (Vol. 31) (pp. 301–349). San Diego: Academic Press.

Schyns, P. G., & Rodet, L. (1997). Categorization creates fuctional features. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 23*(3), 681–696.

Shams, L. (1999). *Development of visual shape primitives*. Unpublished Ph.D. Thesis, University of Southern California, Los Angeles.

Shams, L., & von der Malsburg, C. (in press). The role of complex cells in object recognition. *Vision Research*, in press.

Shams, L. B., Brady, M. J., & Schaal, S. (2001). Graph matching vs mutual information maximization for object detection. *Neural Networks, 14*, 345–354.

Solina, F., & Bajcsy, R. (1990). Recovery of parametric models from range images: the case for superquadrics with global deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(2), 131–147.

Terzopoulos, D., Witkin, A., & Kass, M. (1988). Symmetry-seeking models for 3D object reconstruction. *International Journal of Computational Vision, 1*(3), 1988.

von der Malsburg, C. (1988). Pattern recognition by labeled graph matching. *Neural Networks, 1*, 141–148.

Weber, M., Welling, M., & Perona, P. (2000). *Unsupervised learning of models for recognition*. Paper presented at the 6th European Conference Computational Vision, ECCV2000, Dublin, Ireland, June 2000.

Wiskott, L., Fellous, J.-M., Krüger, N., & von der Malsburg, C. (1995). *Face recognition and gender determination*. Paper presented at the International Workshop on Automatic Face- and Gesture-Recognition, Zurich, June 26–28.

Wiskott, L., Fellous, J.-M., Krüger, N., & von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 775–779.

Wiskott, L., & von der Malsburg, C. (1993). A neural system for the recognition of partially occluded objects in cluttered scenes. *International Journal of Pattern Recognition and Artificial Intelligence, 7*(4), 935–948.

Wiskott, L., & von der Malsburg, C. (1995). *Recognizing faces by dynamic link matching*. Paper presented at the International Conference on Artificial Neural Networks, Paris, France.

Zerroug, M., & Nevatia, R. (1993). *Quasi-invariant properties and 3-D shape recovery of non-straight, non-constant generalized cylinders*. Paper presented at the Proceedings of Computer Vision and Pattern Recognition, New York.

Zerroug, M., & Nevatia, R. (1996). Volumetric descriptions from a single intensity image. *International Journal of Computer Vision, 20*(1–2), 11–42.