MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 1391 December, 1992

Recognition by Prototypes

Ronen Basri

Abstract

A scheme for recognizing $3D$ objects from single $2D$ images is introduced. The scheme proceeds in two stages. In the first stage, the *categorization stage*, the image is compared to prototype objects. For each prototype, the view that most resembles the image is recovered, and, if the view is found to be similar to the image, the class identity of the object is determined. In the second stage, the *identification stage*, the observed object is compared to the individual models of its class, where classes are expected to contain objects with relatively similar shapes. For each model, a view that matches the image is sought. If such a view is found, the object's specific identity is determined. The advantage of categorizing the object before it is identified is twofold. First, the image is compared to a smaller number of models, since only models that belong to the object's class need to be considered. Second, the cost of comparing the image to each model in a class is very low, because correspondence is computed once for the whole class. More specifically, the correspondence and object pose computed in the categorization stage to align the prototype with the image are reused in the identification stage to align the individual models with the image. As a result, identification is reduced to a series of simple template comparisons. The paper concludes with an algorithm for constructing optimal prototypes for classes of objects.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology and the McDonnell-Pew Center for Cognitive Neuroscience. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-91-J-4038. Ronen Basri is supported by the McDonnell-Pew and the Rothchild postdoctoral fellowships.

1 Introduction

Our world contains an overwhelming variety of objects. While people demonstrate outstanding abilities to memorize and recognize thousands of objects [27, 37, 38], computer vision applications largely fail to accommodate these numbers. Apparently, the main tool that enables people to effectively handle this massive amount of objects is categorization. By dividing the objects into classes, the visual system is capable of concluding properties of unfamiliar objects from their resemblance to familiar ones. For familiar objects, categorization offers an indexing tool into the stored library of object representations.

Recognition can be performed in different "levels of abstraction". For example, the same object can be recognized as a face, a human face, or as a specific person's face. Psychological studies suggest the existence of a preferred level for recognition, called "the basic level of abstraction" [33]. Existing computational schemes usually approach recognition in either one of two levels. Several schemes attempt to classify objects in their basic level of abstraction (we refer to this task by *categorization*), while other schemes attempt to determine the specific identity of objects (we refer to this task by *identification*). This paper presents a novel approach for recognition that combines the two tasks.

To see how the two tasks are related, consider the following example. Suppose you are walking down a street, and someone is coming towards you. You look at the person's face, and it looks familiar, but you cannot tell who it is. So you try to picture the people you know who look like the person you see, until finally, you realize who the person is.

A number of hypotheses can be drawn from this story. First, recognition can be broken into two stages: categorization and identification, where categorization is believed to precede identification. Second, during the course of recognition the image is compared against a number of object models. Assuming that indeed categorization precedes identification, only models that belong to the object's class need to be considered. Finally, when a new model is compared to the image, the comparison process may benefit from the use of information acquired during categorization. Note that the situation described here is not specific to faces. One can imagine that similar situations occur when other objects, such as animals, cars, and chairs, are observed.

To see how information acquired during categorization can be used for identification, consider the example of face recognition. When a face is recognized, the image positions of its parts and features are known. In particular, an observer already knows where the eyes, nose, and mouth are and can even infer the direction of gaze and expression. The person's identity is not essential for extracting and locating these features. Instead, they are matched against features in a "generic" representation. In addition, other features, such as a beard, hair style, and wrinkles, that may better distinguish between different persons may be located. More generally, we can postulate that, during categorization, sub-structures of the objects (such as parts and features) are extracted and located with respect to a generic model, and the object's pose is determined.

To follow this example, I propose a scheme for recognizing $3D$ objects from single $2D$ views that combines the two stages, categorization and identification. Categorization is achieved by aligning the image to prototype objects. The prototype that appears most similar to the image determines the class identity of the object. After the object is categorized, its specific identity is determined by aligning the observed object to individual models of its class. By first categorizing the object, not only the number of models considered for identification is reduced, but also the cost of comparing each model to the image significantly decreases. This is achieved by reusing the correspondence and pose computed for the prototype in the categorization stage to align the image with the individual models. We show in this paper that, albeit a perfect match between the prototype and the image is not obtainable, the correspondence and pose can be computed for the prototype, and can be used to bring the image and the object's model into alignment. Consequently, recovering the correspondence and pose for the individual models becomes unnecessary, and identification is reduced to a series of simple template comparisons.

The rest of this paper is divided as follows. Section 2 reviews the main existing approaches for categorization and identification. Section 3 presents the scheme of recognition by prototypes. Section 4 proposes an algorithm for generating optimal prototypes for the scheme. Section 5 discusses the relevance of the scheme to human recognition. Implementation results are presented in Section 6.

2 Previous Approaches

Existing schemes for categorization often use a "reductionist" approach. The image, which contains a detailed appearance of an object, is transformed into a compact representation that is invariant for all objects of the same class. One common approach to generating such a representation is by decomposing the object into parts. Parts are extracted by cutting the object in concavities [17, 22, 43] and labeled according to their general shape. The labels, together with the spatial relationships between the parts, are used to identify the class of the object [4, 6, 7, 26]. A second approach extracts the parts of the object that fulfill certain functions. The list of functions is used to determine the object's class [16, 39, 47].

Schemes that break objects into parts are insufficient to explain all the aspects of recognition for the following reasons. First, in many cases objects that belong to the same class differ only by their detailed shape, while they share roughly the same set of parts. Moreover, even objects that at some level may be considered belonging to different classes, such as a cat and a dog, may also share roughly the same set of parts. To solve this problem several systems also store, in addition to the part structure of the objects, the detailed shape of the parts [2, 6, 7]. Another problem is that many of the techniques for recognizing objects by part decomposition rely on finding the entire parts from the image.

To recognize the specific identity of objects, a relatively detailed representation of the object's shape is compared with the image. An example for such methods is alignment [3, 9, 12, 13, 18, 25, 40, 41]. Alignment involves recovering the position and orientation (*pose*) in which the object is observed and comparing the appearance of the object from that pose with the image. Only a few attempts have been made in the past to extend the alignment scheme to the problem of object categorization (e.g., [36]). The main difficulty in applying the alignment approach is the recovery of the pose of the observed object. In most implementations this involves a time-consuming stage for finding the correspondence between the model and the image. The process becomes impractical when the image is compared against a large library of objects, because typically the correspondence is established between the image and each of the models in the library separately.

To handle large libraries, indexing methods were proposed (e.g., [20, 46, 14]). The basic idea is the following. A certain function is defined and applied to the views of all the objects in the library. The object models are arranged in a look-up table indexed by the obtained function values. When an image is given, the function is applied to the image, and the obtained value is used to index into the table. To reduce the size of the table and the complexity of its preparation, invariant functions, functions that when applied to different views of an object return the same value regardless of viewpoint, often are used as the indexing functions.

Indexing methods suffer from several shortcomings. First, existing indexing methods handle only rigid objects. Extending these methods to handle classes of objects has not been discussed. Second, because of complexity issues, indexing functions usually are applied to small numbers of features. As a result, high rates of false positives are obtained, and the effectiveness of the indexing is reduced.

The scheme presented in this paper is designed to work where traditional approaches to categorization and indexing fail. The scheme combines both categorization and identification of objects, and uses fairly detailed representations for objects. Rather than indexing directly to the specific object model, the scheme indexes into the library of objects by categorizing the object. The classes handled by the scheme include objects with relatively similar shapes. To fit into the scheme, in some cases basic level classes are broken into sub-classes. The general problem of categorization therefore may require additional tools.

3 Recognition by Prototypes

The recognition by prototypes scheme proceeds as follows. A library of $3D$ object models is stored in memory. The models in the library are divided into classes, and $3D$ prototype objects are selected to represent the classes. For every class, the models in the class are aligned in the library with the prototype object. The role of this $3D$ alignment will become clear shortly.

At recognition time, an incoming $2D$ image is first matched against all of the prototypes. For each prototype object, the system attempts to recover the view of the prototype that most resembles the image. To do so, the system recovers the correspondence between the prototype and the image, and, using this correspondence, it determines the transformation that best aligns the prototype with the image. This transformation, referred to as the *prototype transform*, is then applied to the prototype, and the similarity between the transformed prototype and the actual image is evaluated. Since the observed object in general differs from the prototype object, a perfect match between the two is not anticipated. The system therefore seeks a prototype that reasonably matches the image. Once such a prototype is found, the class identity of the object is determined.

After the object's class is determined, the system attempts to recover the specific identity of the object. At this stage, the image is matched against all the models of the object's class. For each of these models, the system seeks to recover the transformation that aligns the model with the image. As will be shown below, since the models are aligned in the library with the prototype, the transformation that best aligns the prototype with the image is identical to the transformation that aligns the model to the image. The prototype transform therefore is applied to the specific models, and their appearance from this pose is compared with the image. The model that aligns with the image, if there is such, determines the specific identity of the object.

The rest of this section is divided as follows. In Section 3.1 the object representation used in our scheme is presented. Section 3.2 describes the categorization stage, and Section 3.3 describes the identification stage.

3.1 Object representation – the linear combination scheme

In our scheme, an object is modeled by a matrix M of size $n \times k$, where n is the number of feature points, and k represents the degrees of freedom of the object. A vector $\vec{a} \in \mathcal{R}^k$, referred to as the *transform vector*, represents the transformation applied to the object in a certain view, and the object's appearance from this view is given by

$$\vec{v} = M\vec{a} \tag{1}$$

In the rest of this section we explain the use of this notation. The notation follows from the linear combination scheme [42], which is briefly reviewed below.

Under the linear combination scheme an object is modeled by a small set of views, each is represented by a vector containing point positions, where the points in these views are ordered in correspondence. Novel views of the object are obtained by applying linear combinations to the stored views. Additional constraints may apply to the coefficients of this linear combination. Computing the object pose therefore requires recovering the coefficients of the linear combination that align the model with the image and verifying that the recovered coefficients indeed satisfy the constraints. The method handles rigid objects under weak-perspective projection (namely, orthographic projection followed by a uniform scaling). It was extended to approximate the appearance of objects with smooth bounding surfaces and to handle articulated objects. In our representation, the columns of the model matrix M contain views of the object, and the coefficients of the linear combination that align the model with the image are given by the transform vector $\vec{a}$.

For concreteness, we review the linear combination scheme for rigid objects. Consider a $3D$ object O that contains n feature points (X_i, Y_i, Z_i), $1 \le i \le n$. Under weak-perspective projection, the position of the object following a rotation R, translation $\vec{t}$, and scaling s is given by

$$\begin{aligned} x_i &= s\, r_{11} X_i + s\, r_{12} Y_i + s\, r_{13} Z_i + t_x \\ y_i &= s\, r_{21} X_i + s\, r_{22} Y_i + s\, r_{23} Z_i + t_y \end{aligned} \quad (2)$$

where r_{ij} are the components of the rotation matrix, R, and t_x, t_y are the horizontal and vertical components of the translation vector, $\vec{t}$ respectively.

Denote by $\vec{X}, \vec{Y}, \vec{Z}, \vec{x}, \vec{y} \in \mathcal{R}^n$ vectors of X_i, Y_i, Z_i, x_i and y_i values respectively, and denote $\vec{1} = (1, \ldots, 1) \in \mathcal{R}^n$, we can rewrite Eq. 2 in a vector equation as follows:

$$\begin{aligned} \vec{x} &= a_1 \vec{X} + a_2 \vec{Y} + a_3 \vec{Z} + a_4 \vec{1} \\ \vec{y} &= b_1 \vec{X} + b_2 \vec{Y} + b_3 \vec{Z} + b_4 \vec{1} \end{aligned} \quad (3)$$

where

$$\begin{aligned} a_1 &= s\, r_{11} & b_1 &= s\, r_{21} \\ a_2 &= s\, r_{12} & b_2 &= s\, r_{22} \\ a_3 &= s\, r_{13} & b_3 &= s\, r_{23} \\ a_4 &= t_x & b_4 &= t_y \end{aligned}$$

Therefore

$$\vec{x}, \vec{y} \in span\{\vec{X}, \vec{Y}, \vec{Z}, \vec{1}\} \quad (4)$$

Different views of the object are obtained by changing the rotation, scale, and translation parameters, and these changes result in changing the coefficients in Eq. 3. We may therefore conclude that all the views of a rigid object are contained in a $4D$ linear space.

This property, that the views of a rigid object are contained in a $4D$ linear space, provides a method for constructing viewer-centered representations for the object. The idea is to use images of the object to construct a basis for this space. In general, two views provide sufficiently many vectors. Therefore, any novel view is a linear combination of two views [30, 42].

Not every linear combination is a valid view of a rigid object. Following the orthonormality of the row vectors of the rotation matrix, the coefficients in Eq. 3 must satisfy the two quadratic constraints

$$\begin{aligned} a_1^2 + a_2^2 + a_3^2 &= b_1^2 + b_2^2 + b_3^2 \\ a_1 b_1 + a_2 b_2 + a_3 b_3 &= 0 \end{aligned} \quad (5)$$

When the constraints are not satisfied, distorted (by stretch or shear) pictures of the objects are generated. In case a viewer-centered representation is used, the constraints change in accordance with the selected basis. A third view of the object can be used to recover the new constraints.

For the purpose of this paper a model for a rigid object can be constructed by building the following $n \times 4$ model matrix

$$M = (\vec{X}, \vec{Y}, \vec{Z}, \vec{1})$$

Views of the object can be constructed as follows

$$\begin{aligned} \vec{x} &= M\vec{a} \\ \vec{y} &= M\vec{b} \end{aligned} \quad (6)$$

where $\vec{a} = (a_1, a_2, a_3, a_4)$ and $\vec{b} = (b_1, b_2, b_3, b_4)$ are the coefficients from Eq. 3. Notice that the two linear systems can be merged into one by constructing a modified model matrix in the following way

$$\begin{pmatrix} \vec{x} \\ \vec{y} \end{pmatrix} = \begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix} \begin{pmatrix} \vec{a} \\ \vec{b} \end{pmatrix} \quad (7)$$

Similar constructions can be obtained for objects with smooth bounding surfaces and for articulated objects. The width of M, k, should then be modified according to the degrees of freedom of the modeled object. As was mentioned above, viewer-centered representations can be obtained by constructing a basis for the $4D$ space from images of the object. Therefore, viewer-centered models can be obtained by replacing the column vectors of M with the constructed basis.

To summarize, following the linear combination scheme we can represent an object by a matrix M and construct views of the object by applying it to transform vectors $\vec{a}$. For rigid objects not every transform vector is valid; the components of the transform vector must satisfy the two quadratic constraints. Recognition involves recovering the transform vector $\vec{a}$ and verifying that its components satisfy the two constraints. Ignoring these constraints will result in recognizing the object even when it undergoes general $3D$ affine transformation. In the analysis below we largely ignore the quadratic constraints. These constraints, however, can be verified both during the categorization stage as well as during the identification stage.

3.2 Categorization

The recognition by prototypes scheme begins by determining the object's category. This is achieved by comparing the observed object to prototype objects, objects that are "typical exemplars" for their classes. For a given prototype, the view of the prototype that most resembles the image is recovered and compared to the actual image, and the result of this comparison determines the class identity of the object.

We begin our description of the categorization stage by defining the data structures used by the scheme. A class $\mathcal{C} = (P, \{M_1, M_2, \ldots, M_l\})$ is a pair that includes a prototype P and a set of object models $M_1, M_2, \ldots, M_l$. Both the prototype and the models are represented by $n \times k$ matrices, where n defines the number of feature points considered, and k denotes the degrees of freedom of the objects. For the sake of simplicity we assume here that all the objects in the class share the same number of feature points, n, and that they have similar degrees of freedom, k. Note that similar objects tend to have similar degrees of freedom (e.g., all of them are rigid). Both assumptions are not strict, however. The scheme can be modified to tolerate both varying number of feature points as well as different degrees of freedom. The details will be discussed later in this paper. Note that the objects can be modeled by either object-centered or viewer-centered representations. In case viewer-centered representations are used we shall assume that the models represent the objects from the same range of viewpoints.

A class in our scheme contains objects with similar shapes. These objects share roughly the same topologies, and there exists a "natural" correspondence between them. Consider, for instance, the two chairs in Figure 1. Although the shapes of these chairs are different, and some parts (e.g., the arms) appear only in one chair and not in the other, a natural correspondence between features in the two objects can be determined.

In the library of models, the natural correspondence between objects is made explicit. It is specified by the order of the row vectors of the models. Specifically, given a prototype P and object models $M_1, \ldots, M_l$, we order the rows of these models such that the first feature point of P corresponds to the first feature point of each of the models $M_1, \ldots, M_l$, and so forth.

Given the library of objects and given an incoming image, the recognition by prototypes scheme begins by categorizing the object observed in the image. To achieve this goal, the prototype objects are aligned and compared to the image. For every prototype, the correspondence between the image and the prototype is first resolved, and, using this correspondence, the nearest prototype view is recovered. By doing so, the scheme decouples the two factors that affect the appearance of the object in the image, namely, view variations and shape variations. By selecting the nearest prototype view to the image, the scheme compensates for view variations. Then, by evaluating the similarity between the nearest prototype view and the actual image, it accounts for the differences in shape between the prototype and the observed object.

The first stage in matching the prototype to the image involves the recovery of correspondence between prototype and image features. In existing systems for recognizing the specific identity of objects establishing the correspondence between images and object models involves a time-consuming process in which sophisticated algorithms are applied [10, 13, 15, 18, 23, 25, 35, 41]. These algorithms rely on the property that, when the correct correspondence between a model and an image is established, a near-perfect match between the two is obtained. While this assumption is valid for identification, it cannot be used under our scheme since the prototype and the image generally represent different objects.

To determine the correspondence between the prototype and the image, we define an objective function that is applied to the prototype and the image under a given correspondence and that obtains its minimum under the correct correspondence. The objective function will measure the quality of the match between the prototype and the image. Namely, under this measure the correct correspondence is the one that brings the prototype into its best alignment with the image. Given this objective function, correspondence is a combinatorial optimization problem, and so minimization techniques can be used to resolve the correspondence between the prototype and the image. This paper does not propose a specific technique for solving the correspondence problem.

Assuming the correspondence problem can be solved, the scheme proceeds as follows. Given a prototype P and an image I, we generate a view vector $\vec{v}$ from the image by extracting the location of feature points and arranging them in a vector. The points in $\vec{v}$ are ordered in correspondence to the prototype points; that is, the first point in $\vec{v}$ corresponds to the first point in P and so forth. The *prototype transform* is the transformation that brings the prototype points as close as possible to their corresponding image points. The prototype transform, therefore, is the transform vector $\vec{b}$ that minimizes the Euclidean distance between the prototype and image points, namely

$$\min_{\vec{b}'} \|P\vec{b}' - \vec{v}\| \tag{8}$$

A solution for (8) is obtained as follows. Assuming P is overdetermined; that is, P is $n \times k$ where $n > k$ and $rank(P) = k$, and denote by $P^+ = (P^TP)^{-1}P^T$ the pseudo-inverse of P, the prototype transform, $\vec{b}$, is given by

$$\vec{b} = P^+\vec{v} \tag{9}$$

and the *nearest prototype view* $\vec{p}$ is obtained by applying P to the prototype transform, $\vec{b}$, that is

$$\vec{p} = P\vec{b} = PP^+\vec{v} \tag{10}$$

The nearest prototype view is now compared to the image, and their resemblance determines the class identity of the object. The quality of the match between the prototype and the image is defined by

$$D(P, \vec{v}) = \|\vec{p} - \vec{v}\| = \|(I - PP^+)\vec{v}\| \tag{11}$$

To eliminate effects due to scaling of the object, this measure should be normalized, as is illustrated by the example below. Consider an object seen from some view $\vec{v}_1$. Its distance to the prototype is given by $D(P, \vec{v}_1)$. Suppose the object is now seen from a new view $\vec{v}_2$ that is identical to $\vec{v}_1$ except that the object is now as twice as close to the camera. Under these conditions $\vec{v}_2 = 2\vec{v}_1$, and its distance to the prototype is given by $D(P, \vec{v}_2) = 2D(P, \vec{v}_1)$. Clearly, we should have a measure that is independent of the distance of the object to the camera. One way to obtain such a measure is by dividing $D(P, \vec{v})$ by the norm $\|\vec{v}\|$

$$\hat{D}(P, \vec{v}) = \frac{\|(PP^+ - I)\vec{v}\|}{\|\vec{v}\|} \tag{12}$$

$\hat{D}(P, \vec{v})$ is proposed here as an objective function for establishing the correspondence between the prototype and the image. In other words, we expect that if the object belongs to the prototype's class then $\hat{D}(P, \vec{v})$ obtains its minimal value when $\vec{v}$ is ordered in correspondence to P. Any other permutation will increase the value of $\hat{D}$. Formally, denote by σ a permutation matrix, we assume that

$$\hat{D}(P, \vec{v}) = \min_{\sigma} \hat{D}(P, \sigma\vec{v}) \tag{13}$$

The measure $\hat{D}(P, \vec{v})$ has a second role. Since it measures the similarity between the prototype and the image, it can also be used to determine the object's class.

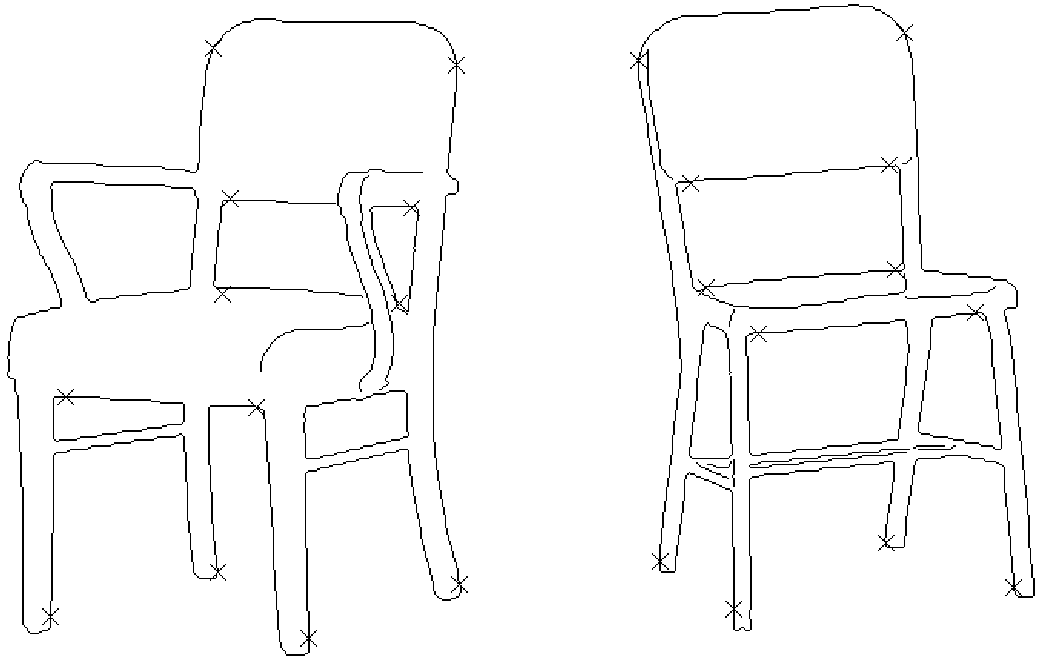

Figure 1: "Natural" correspondences between two chairs

An object observed in a view $\vec{v}$ belongs to the class represented by a prototype P if

$$\hat{D}(P, \vec{v}) < \epsilon \quad (14)$$

for some constant $\epsilon > 0$. We refer to (14) as the *categorization criterion*.

The categorization stage proceeds as follows. Given an image I and a prototype P, the correspondence between P and I is resolved by minimizing the measure $\hat{D}(P, \sigma\vec{v})$ over all possible permutation σ of $\vec{v}$, and if the obtained minimum $\hat{D}(P, \vec{v})$ is below the threshold ϵ, then the class identity of the object is determined.

Note that in our scheme the prototype and the categorization criterion determine the actual division of objects to classes; an object belongs to a certain class if its views are sufficiently similar, according to the categorization criterion, to views of the prototype. Under the above definition, an object belongs to a prototype's class if the total difference between its feature points and their corresponding prototype points does not exceed ϵ.

The measure $\hat{D}(P, \vec{v})$ defined here determines the similarity between the prototype P and the view $\vec{v}$ using only the distances between feature points. In general, since correspondence is difficult to achieve, such a measure would not be robust. Including additional information about the features in the similarity measure may increase the robustness of the scheme. Also, measures that consider only the proximity of feature points are limited in terms of dividing the library into classes, since they induce classes of objects with highly similar shapes. Measures that consider additional information can extend the classes to include larger sets of objects.

The measure $\hat{D}(P, \vec{v})$ can be enriched by considering the similarity between corresponding points. A simple example for a measure that considers both the proximity and similarity between feature points is the following measure. Each feature point is associated with a label (such as a corner or an inflection point). Again, the measure $\hat{D}(P, \vec{v})$ is applied, but this time only correspondences between points with similar labels are allowed; namely, corners in the image can only match corners in the prototype, and, similarly, inflection points can only match inflection points. Other examples for measures that combine proximity and similarity include measures that retain the tangent or the curvature of points. More sophisticated measures may compare the topologies of the objects in the two views, or, in other words, verify that the objects share similar part structures in $2D$.

A useful technique in measuring the similarity between the image and the nearest prototype view is to consider a different set of features than the set used to determine the prototype transform. The rational behind this technique is that it is generally difficult to recover exact feature-to-feature correspondence, and while such correspondences are necessary for recovering the prototype transform, similarity measures can be successfully applied even in the absence of exact feature-to-feature correspondence. This idea resembles the basic principle of the alignment algorithm[18, 41], in which a small set of points is used to compute the object pose, while a larger set of points is used to verify this pose.

It should be noted that the general flow of the scheme and, in particular, the identification stage are independent of the specific choice of similarity measure. As has been noted above, the measure affects the division of model libraries into classes and the selection of optimal prototypes for these classes. An example for selecting the optimal prototype for a given class under the measure specified in (12) (for either labeled or unlabeled features) is described in Section 4.

Finally, although the main objective of the categorization stage is to determine the class identity of the object, the categorization scheme described above is useful even if the object's category cannot be determined. Section 3.3 below shows that the prototype transform can be reused to align the image with the specific models. Consequently, following the categorization stage the cost of comparing the image to each of the specific models is substantially reduced since the difficult part of recovering the transformation that relates the models to the image is applied only to the prototype objects. As a result, if the class identity of the object cannot be determined we still need to consider all the specific models in the library, but the overall cost of comparing the models to the image would be low because correspondence is computed once for the whole class.

3.3 Identification

After the observed object is categorized, the system turns to recovering its individual identity. At this stage the image is matched to all the models in the object's class. For each model, the system seeks to recover the transformation that aligns the model to the image, if there is such. In previous schemes this required recovering the correspondence between the image and each of the models separately. In our scheme, however, this no longer is necessary, since the object transform is determined directly from the prototype transform. We show in this section that the prototype and the object transforms are related by a simple transformation, which can be computed in advance, and which can in fact be undone already in the library of stored models. Consequently, the prototype transform can be reused in the identification stage to align the individual models with the image.

The initial stage of categorization recovers three pieces of information that can be used for identification. The three are (i) the object class, (ii) the correspondence between the prototype and the image, and (iii) the prototype transform. This information is used in the identification stage as follows. First, since the object's class is determined, only models that belong to this class are considered. Second, using the correspondence between the prototype and the image established in the categorization stage, and using the stored correspondence between the prototype and the object models, the correspondence between the models and the image is immediately recovered. Finally, as is shown below, the model transform, namely, the transformation that aligns the model with the image, is recovered from the prototype transform.

Assume we are given with a view $\vec{v}$ of some object model M_i, namely

$$\vec{v} = M_i \vec{a} \qquad (15)$$

for some transform vector $\vec{a}$. When the identification process begins, it is still unknown which of the models $M_1, ..., M_l$ of the object's class accounts for the image and what the transform vector $\vec{a}$ is. The first task faced by the scheme at this stage is to recover the model transform, $\vec{a}$. This is done, as is explained below, using the prototype transform $\vec{b} = P^+ \vec{v}$ defined in (9). Once $\vec{a}$ is recovered, it is applied to all the models $M_1, ..., M_l$, and the model for which a near-perfect match is obtained determines the object's identity.

Theorem 1 below establishes that the model transform $\vec{a}$ can be recovered directly from the prototype transform $\vec{b}$ by applying a linear transformation which is referred to as the *prototype-to-model transform*. This transform has two interesting properties. First, it is view-independent; namely, for any given view of the object, the same transform maps the prototype transform that corresponds to this view to the correct model transform. The prototype-to-model transform therefore can be computed in advance and stored in the library of models. Second, the prototype-to-model transform can be used to recover the model transform regardless of the quality of match between the prototype and the image. In other words, even if the prototype aligns poorly with the image, the transformation that aligns the model with the image is determined correctly in this process.

Theorem 1: *Given a view $\vec{v} = M_i \vec{a}$. Let $\vec{b} = P^+ \vec{v}$ be the prototype transform, that is, the transform vector that best aligns the prototype with the image. The model transform, $\vec{a}$, can be recovered from the prototype transform, $\vec{b}$, by applying a matrix, A_i, namely*

$$\vec{a} = A_i \vec{b}$$

A_i is referred to as the prototype-to-model transform

Proof: Notice that

$$\vec{b} = P^+ \vec{v} = P^+ M_i \vec{a}$$

Assume $P^+ M_i$ is invertible, let

$$A_i = (P^+ M_i)^{-1}$$

we obtain that

$$\vec{a} = A_i \vec{b}$$

□

Corollary 2: *The prototype-to-model transform is view-independent.*

Proof: The prototype-to-model transform, A_i, is independent of both pose vectors, $\vec{a}$ and $\vec{b}$. Changing the image $\vec{v}$ will result in a new pair of pose vectors, $\vec{a}$ and $\vec{b}$, but similar to the old pair, the new pair is related through the same transform A_i. The prototype-to-model transform A_i therefore can be used to recover the object pose for any view of M_i. □

A_i exists if $P^+ M_i$ is invertible. This condition is equivalent to requiring that the two column spaces of P and M_i will not be orthogonal in any direction. The condition holds, in general, when the two objects are fairly similar. This is illustrated by the following example. Consider the case that both column spaces of P and M_i are one-dimensional; namely, each represents a line through the origin. The only case in this one-dimensional example in which A_i does not exist is when P and M_i are orthogonal. But these lines are farthest apart when they are orthogonal. Consequently, if the objects are relatively similar A_i would exist.

Since it depends only on the prototype P and the model M_i, the prototype-to-model transform A_i can be pre-computed and stored in the library of models. Every model $M_i \in \mathcal{C}$ is associated with its own transform A_i that relates, for every possible view of M_i, between the prototype transform and the model transform. To compare the image to the model M_i the model transform should first be recovered. This is achieved by applying A_i to the prototype transform computed in the categorization stage.

Also, the prototype-to-model transform, A_i, can be used to align the model M_i with the prototype P in $3D$. Denote the aligned model by M_i', M_i' models the same object as M does, since their column vectors span the same space. In addition, the aligned model M_i' has the property that it is brought by the prototype transform, $\vec{b}$, to a perfect alignment with the image. Consequently, if the models are aligned in the library with the prototype, the prototype transform computed in the categorization stage can be reused for identification with no further manipulations. This is established in Theorem 3 below.

Theorem 3: *Let $M_i' = M_i A_i$ be the model M_i aligned with the prototype P. For any view $\vec{v} = M_i \vec{a}$ the prototype transform for this view $\vec{b} = P^+ \vec{v}$ is identical to the model transform for this view, that is, $\vec{v} = M_i' \vec{b}$.*

Proof: Since

$$M_i' = M_i A_i$$

we obtain that

$$M_i' \vec{b} = M_i A_i \vec{b} = M_i \vec{a} = \vec{v}$$

□

Using Theorem 3, the identification scheme is simplified as follows. The models $M_1, \ldots, M_l$ are aligned in the library with the prototype P by applying the corresponding prototype-to-model transform, $A_1, \ldots, A_l$. At recognition time, the prototype transform $\vec{b} = P^+ \vec{v}$, is applied to the aligned models $M_1', \ldots, M_l'$. According to Theorems 1 and 3, by transforming the models by $\vec{b}$ the correct model, M_i', would perfectly align with the image.

In the scheme above we assumed that full feature-to-feature correspondence is established between the prototype and the image. This assumption is not mandatory. Methods for estimating the prototype transform using partial correspondence or by considering other types of features (such as line segments) can also be used. Note that in case the prototype transform can only be approximated, the accuracy of the model transform obtained is determined by the quality of this approximation and by the condition number of the prototype-to-model transform A_i. The condition number of A_i affects the match even if Theorem 3 is applied, namely, even if the models are aligned with the prototype in advance. Consequently, the condition number of the prototype-to-model transform A_i should be taken into account when the library is divided into classes.

Finally, the scheme can be extended to handle classes of objects with different degrees of freedom. Consider, for instance, the case of similar chairs, some of which are folding. Obviously, the folding chairs have more degrees of freedom than the regular, rigid chairs, and therefore they would be represented in the library by wider matrices than the rigid chairs are. As is explained below, the chairs can be handled in a common class, and the prototype for the class would itself be a folding chair.

More generally, let $M_1, \ldots, M_l$ be a class of models of different widths, and denote by $k_1, \ldots, k_l$ the width of $M_1, \ldots, M_l$ respectively. Let P be the prototype for this class, and denote by k_p the width of P, we set k_p to be

$$k_p = \max\{k_1, \ldots, k_l\} \tag{16}$$

In other words, we require the prototype to have the same degrees of freedom as the most flexible object in the class. We can set k_p according to our goal since, as it is shown in Section 4, the prototype P is obtained in our scheme by manipulating the objects in the class. The prototype-to-model transform A_i is defined in this case by

$$A_i = (P^+ M_i)^+ \tag{17}$$

where A_i is $k_p \times k_i$. It is straightforward to extend Theorem 1 to also include this case. Consequently, for any view of M_i, the model transform $\vec{a}$ can be recovered from its corresponding prototype transform $\vec{b}$ by applying the prototype-to-model transform A_i to $\vec{b}$. Note that since $k_p \geq k_i$ the prototype can appear in poses that do not match any possible model pose (and therefore in noiseless conditions they are impossible to obtain). In case the object is observed from such a view, A_i would map this unmatched prototype transform to the model transform that corresponds to the nearest matched prototype transform. By setting k_p to be as large as the maximum of $k_1, \ldots, k_l$ we avoid cases where there exist views of the object that cannot be accounted for by the prototype. Model transforms that correspond to such views cannot be recovered from prototype transforms.

3.4 Summary

We presented in this section a scheme for recognizing $3D$ objects from single $2D$ views that proceeds in two stages, categorization and identification. In the categorization stage the image is compared against the stored prototypes. For every prototype, the correspondence between the image and the prototype is recovered, and the nearest view of the prototype is constructed. The similarity between this view and the image is evaluated, and, if the two are found similar, the class identity of the object is determined. In the identification stage the observed object is compared against the models of its class. Since the prototype and the models were brought in the library into alignment, the same transformation that aligns the prototype to the image also aligns the object model to the image. The prototype transform therefore is applied to the models, and the obtained views are compared with the image. The view that is found to be identical up to noise and occlusion to the image determines the individual identity of the object.

The presented scheme is based on several key principals. Recognition is divided into two sub-processes, categorization and identification. In both processes models are aligned with the image, and the identity of the object is determined by a $2D$ comparison; $3D$ reconstruction of the observed object from the image is not performed. The difficult component of the alignment approach, namely, the recovery of correspondence and object pose, is performed only once for each class; the prototype transform is reused in the identification stage to align the image with the individual models.

4 Constructing optimal prototypes

In the scheme above we assumed that the classes in the library of models are represented by prototype objects. Since categorization is achieved by matching the image to prototype objects, the question of how to select the best prototype should be addressed. In this section we present an algorithm for constructing optimal prototypes.

Given a class of objects, the optimal prototype for this class is the object that resembles the objects of the class the most. Under our formulation, such an object would share as many features as possible with the objects of its class, the position of these features on the prototype would be as close as possible to their position on the objects, and the prototype-to-model transform for these objects would be as stable as possible. Below we show that the optimal prototype can effectively be computed using principal component analysis; that is, by computing the dominant eigenvectors for some matrix determined by the models of the class.

Principal component analysis often is used in classification problems to construct classes and prototypes [11]. In existing applications, an object is represented by a point in some high dimensional space, where each component of this point contains an invariant attribute of the object. A hyperplane in that space represents a class of objects. The goal of the principal component analysis is, given a set of points (objects), to recover the class that these points induce. Our case is somewhat different. In our case an object is represented by a continuous linear space rather than by a point. Whereas the use of hyperplanes in other schemes often is arbitrary and made primarily for convenience, their use in our scheme is appropriate following the linear combination scheme [42] (see Section 3.1).

The differences outlined above also imply differences in the proof that principle component analysis applies to our case. We show below that the optimal prototype can be computed by principal component analysis. The traditional proof needs to be extended since in our case objects are represented by continuous spaces rather than by discrete points.

The prototype constructed in this process is a $3D$ object obtained by manipulating the objects in its class. To allow the construction, it seems as if the objects in the class should first be brought into alignment. In particular, if the objects are represented by viewer-centered models (that is, by sets of their views, see Section 3.1 for details), the different objects would then have to be represented by images taken from similar viewpoints. Nevertheless, the process presented below does not require an initial alignment of the objects. The same prototype is obtained in this process even when the objects are not aligned

We now turn to constructing the optimal prototype. First, we define an objective function. Given a prototype P and an object model M_i, we define the similarity between P and M_i as follows. Let $\vec{v}_i$ be a view of M_i, we measure the similarity between the prototype P and the view $\vec{v}_i$ using (12). Then, we sum the measure over all possible views of M_i. Assuming without loss of generality that $\|\vec{v}_i\| = 1$, (14) can be rewritten as

$$\hat{D}(P, \vec{v}_i) = \|(PP^+ - I)\vec{v}_i\| \tag{18}$$

Without loss of generality, we can assume that the constructed prototype, P, is composed of orthonormal columns. Note that an overdetermined matrix P with orthonormal columns satisfies $P^+ = P^T$. We can therefore rewrite (18) as

$$\hat{D}(P, \vec{v}_i) = \|(PP^T - I)\vec{v}_i\| \tag{19}$$

The distance between P and the model M_i is now given by summing $\hat{D}(P, \vec{v}_i)$ over all unit-length (to eliminate scaling effects) views of M_i, namely

$$\hat{D}(P, M_i) = \int_{\|\vec{v}_i\|=1} \|(PP^T - I)\vec{v}_i\| \tag{20}$$

To obtain the objective function, we sum these distances over all models

$$E(P) = \sum_{i=1}^{l} \int_{\|\vec{v}_i\|=1} \|(PP^T - I)\vec{v}_i\| \tag{21}$$

The object P that minimizes this function is defined to be the optimal prototype.

Note that (21) is not the only possible objective function for this purpose. An alternative "worst case" approach is to measure the distance between the prototype to the farthest model in the class (rather than summing this distance over all models). Except for being difficult to compute, this measure also is sensitive to "outlier" models.

The prototype that minimizes (21) can be constructed in a process that includes the following steps.

1. To simplify the process we assume the column vectors of each of the model matrices M_i, $(1 \le i \le l)$, are orthonormal. (In case they are not, we first apply a Gram-Schmidt process to them. Such a process obviously does not alter the space of views implied by the models.)

2. Build the $n \times n$ symmetric matrix

$$F = \sum_{i=1}^{l} M_i M_i^T$$

3. Find the k dominant eigenvectors of F. The optimal matrix P is constructed from these eigenvectors.

Note that, in general, we are trying to construct a prototype object that would belong to the given class. This condition determines the choice of width k for the prototype. If all the models share the same width then the prototype would assume this width. In the rigid case, for example, $k = 4$ (see Section 3.1). As mentioned in Section 3.3 above, in case the objects have different degrees of freedom, k is set to be the maximum of $k_1, \ldots, k_l$, where $k_1, \ldots, k_l$ are the widths of $M_1, \ldots, M_l$ respectively. In case more than k large eigenvalues are obtained, one may ignore these guideline rules and construct a prototype that has higher degrees of freedom than the objects in the class (see for example [31]).

Theorem 4 below establishes that the algorithm above produces the optimal prototype. We consider here the case that all the objects share similar degrees of freedom. The same procedure can be applied with slight modifications to include the case of objects with different degrees of freedom.

Theorem 4: *Let $M_1, M_2, \ldots, M_l$ be a set of models belonging to some class $\mathcal{C}$. Assume every model M_i is represented by an $n \times k$ matrix with orthonormal column vectors. The prototype P that minimizes the term*

$$E(P) = \sum_{i=1}^{l} \int_{\|\vec{v_i}\| = 1} \| (PP^T - I) M_i \vec{v}_i \|$$

where the integration is done over all the unit-length views $\vec{v}_i$ of each model M_i, is composed of the k eigenvectors of the matrix

$$F = \sum_{i=1}^{l} M_i M_i^T$$

that correspond to its k largest eigenvalues.

Proof: Let P be composed of the k dominant eigenvectors of F. According to regression principles P minimizes the term

$$\sum_{i=1}^{l} \sum_{j=1}^{k} \| (PP^T - I) \vec{m}_{ij} \|$$

where $\vec{m}_{ij}$ is the j'th column vector of M_i. In other words, consider $\vec{m}_{ij}$ as a point in $\mathcal{R}^n$. The space spanned by the column vectors of P is the nearest k-dimensional hyperplane to these points, $\vec{m}_{ij}$. The rest of this proof extends the claim from the discrete sum over the column vectors of M_i to the continuous integral over all views spanned by these vectors. According to our assumptions, each matrix M_i contains an orthonormal set of column vectors. Replacing these vectors by another orthonormal basis for M_i will not change the matrix P; that is, P is independent of the choice of orthonormal basis for the models. This is illustrated by the following derivation. To obtain a new orthonormal basis for the column space of M_i we can apply a $k \times k$ rotation matrix R to M_i (namely, $M_i R$). P is the best vector space for the new set as well, since

$$M_i R (M_i R)^T = M_i R R^T M_i^T = M_i I M_i^T = M_i M_i^T$$

F therefore is constant for any choice of orthonormal vectors for $M_1, \ldots, M_n$, and so its dominant eigenvectors represent the best vector space for for any orthonormal representation of the objects. Consequently, P minimizes the objective function regardless of choice of basis for the models, and therefore it also minimizes the required term

$$E(P) = \sum_{i=1}^{l} \int_{\|\vec{v_i}\| = 1} \| (PP^T - I) M_i \vec{v}_i \|$$

$\square$

To summarize, we showed that given a class of object models, the optimal prototype for this class is given by the dominant eigenvectors of the matrix F, which is constructed from the object models. Note that in proving Theorem 4 we showed that the prototype is independent of choice of basis for the models. This implies that, in order to construct the prototype, the object models $M_1, \ldots, M_l$ do not need to first be brought into alignment. The process above guarantees to output the same prototype object even if the models are not aligned.

5 Relevance to human vision

The recognition by prototypes scheme uses the general shape of objects as the cue for recognizing them. As was already mentioned, classes in our scheme contain objects with fairly similar shapes. In contrast, the human visual system recognizes objects using both shape cues as well as many other cues, such as color, texture, motion, and context, and objects are categorized in their basic level of abstraction [33]. Only little is currently known about the underlying processes for recognition used by the visual system. From what is known, in spite of the differences pointed above, the recognition by prototypes scheme seems to be consistent in several key issues with psychological and physiological findings. In this section we briefly review these findings.

The scheme presented in this paper promotes the notion that categorization and identification are performed using similar tools. In both cases view variations first are compensated for, and then a view of either the hypothesized prototype or object model is compared with the image. This is in contrast to methods (such as part decomposition and functional description) that in general handle either categorization or identification, but do not extend to deal with both problems. The available studies in this case are inconclusive. Some evidence seem to indicate that the two processes are handled separately by the visual system. Agnostic and prosopagnostic patients often demonstrate degraded identification abilities, whereas their performance in categorization remains intact. Double dissociation between the two processes, however, has not been found, and so the assumption that the two processes are handled separately in the brain has not been established. In fact, both cells that respond to general faces as well as cells that respond to specific faces where found lying side by side within the same brain area, STS, of the macaque monkey [29]. The vulnerability of the identification process to brain lessions can be explained by that the process requires a relatively large memory to encode the detailed shapes of objects as well as sophisticated image processing mechanisms to recover a detailed description of the observed object from the image (see e.g., [19]).

Another idea proposed here is that categorization involves two stages: a stage of compensating for view variations followed by a stage of $2D$ comparison to account for shape differences. A decoupling of view variation and semantic categorization was suggested by Lissauer [24]. Warrington and Taylor [44, 45] found that patients that suffer from lessions in the posterior lobe of the right hemisphere demonstrate difficulties in categorizing objects from unconventional views, whereas their performance in categorization of objects from conventional views remains intact. Additional evidence for the effect of view variations on categorization performance were found for healthy subjects. Subjects that are asked to name objects respond slower when the objects appear in unconventional views [28]. Also, mental rotation effects, namely, response time that grows linearly with the tilt of the object, were observed in naming tasks of natural objects [21].

Finally, the process of categorization presented here is achieved by comparing the image to prototype objects, and these prototype objects can be constructed by manipulating the familiar objects of the class. Recent studies indicate that response time in naming tasks is typically shorter and error rates are lower when the observed object is similar to the prototype [5]. Similarly, shorter reaction time is obtained when subjects are asked to answer questions of the type "does the object X belong to the class Y?" [34]. Other studies reported that children learn good examples of classes before they learn poor ones [1, 32] and that subjects recall having seen the prototype or average configuration of studied face images even if this configuration was not studied [8].

To summarize, although the presented scheme generally does not recognize objects in their basic level of abstraction, it is consistent with psychological and physiological findings in several key issues including a single approach for the two sub-problems of recognition, categorization and identification, view dependency of the two sub-processes, and the role of prototypes in categorization. The findings discussed here obviously are inconclusive, since psychological and physiological studies including the ones discussed here have more than one possible interpretation.

6 Implementation

To test the ideas presented in the paper, we have implemented the scheme and applied it to several objects. In our implementation, the library of models included two classes. The first (Figure 2) contained two four-legged chairs (denoted by A and B), and the second (Figure 3) included two car models, a VW and a Saab.

To demonstrate categorization, we used chair A as a prototype and matched it to an image of chair B. Correspondences between the prototype and the image were picked manually, and, using these correspondences, the prototype transform was recovered and applied to the prototype. The results of matching the transformed prototype with the image are seen in Figure 4. It can be seen that the transformed prototype (middle figure) assumed the same orientation as the observed object (left figure), and that the match between the two is good considering that the objects have different shapes. Note that in this implementation we allowed the objects to undergo general affine transformations in $3D$, including stretch and shear, and so the match between the prototype and the image was better than if only rigid transformations were allowed. Additional examples using chair B and the two cars as the prototypes are shown in Figures 5-7.

In Figures 8-9 we tried to match the prototypes to the images with wrong correspondences. The results of these matches were significantly worse than when the correct matches were used. This is consistent with the idea discussed in Section 3.2 that the quality of the match can be used as the objective function for resolving the correct correspondence.

Figure 10 shows the results of matching a prototype four-legged chair to a single-legged office chair. It can be seen that the upper portions of the chairs match relatively well, while the legs of the chairs do not find appropriate matches.

Figure 11 shows the result of matching a prototype chair to an image of a Saab car. As an anecdotal example, we matched the hole below the back of the chair to the windshield of the car and the seat to the hood. In general, whatever correspondence is used, the two objects would match poorly relative to matching the prototypes to objects of their class.

Figures 12-13 demonstrate the identification stage. In the library we first aligned the model for chair A with the prototype chair (chair B) using the prototype-to-model transform. Then, an image of chair A was categorized (Figure 5) by matching it to the prototype chair, and the prototype transform was computed. In the next step, the prototype transform was applied to the specific model of chair A. The result of this application is seen in Figure 12. It can be seen that a near-perfect alignment was achieved in this process. A similar process was applied to the VW car in Figure 13 using the Saab car as the prototype. (The result of the corresponding categorization stage is shown in Figure 6.) These figures demonstrate that although a perfect match between the prototype and the image could not be obtained, the prototype transform can still be used to align the observed object with its specific model.

7 Summary

We introduced in this paper a recognition scheme that proceeds in two stages: categorization and identification. Categorization is achieved by aligning the image to prototype objects. For every prototype, the nearest prototype view is recovered, and the similarity between this view and the image is evaluated. The prototype that most resembles the observed object determines its class identity. Likewise, identification is achieved by aligning the observed object to the individual models of its class. At this stage the prototype transform computed in the categorization stage is reused to align the models with the image. The model that matches the observed object determines its specific identity. In addition, we

Figure 2: Pictures of two chairs used as models. We refer to these chairs by A(left) and B(right). Models for the two chairs were constructed from single images using symmetry [31].

Figure 3: Pictures of two cars used as models. Left: a VW model. Right: a Saab model. Models for the two cars were borrowed from [42].

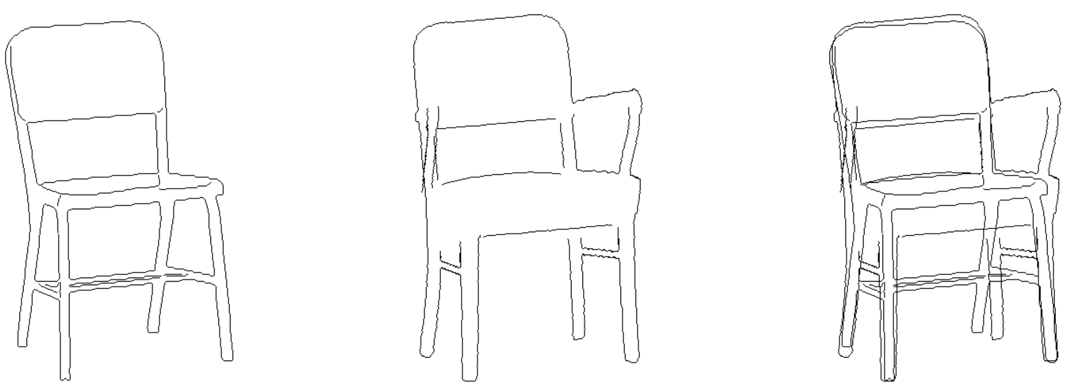

Figure 4: Matching a prototype chair (chair A) to an image of chair B. This figure, as well as the rest of the figures, contain three pictures. Left: the image to be recognized. Middle: the appearance of the prototype following the application of the prototype transform. Right: an overlay of the left and the middle pictures.

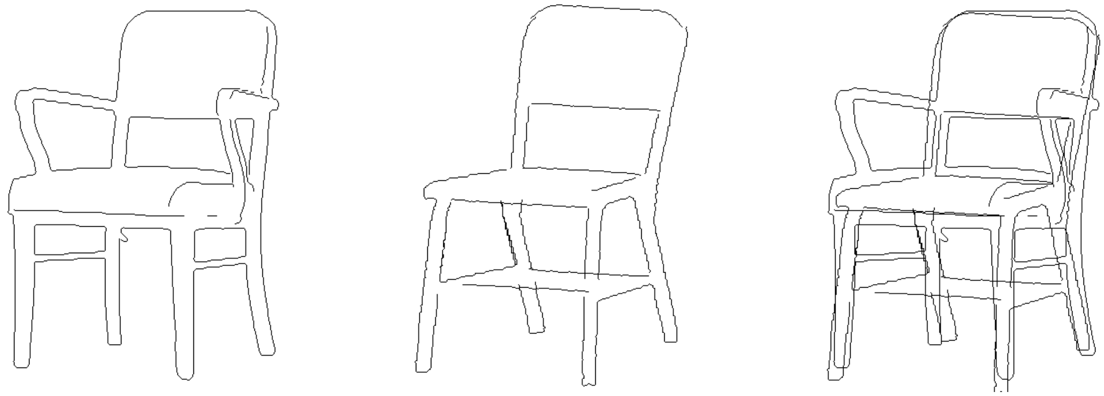

Figure 5: Matching a prototype chair (chair B) to an image of chair A

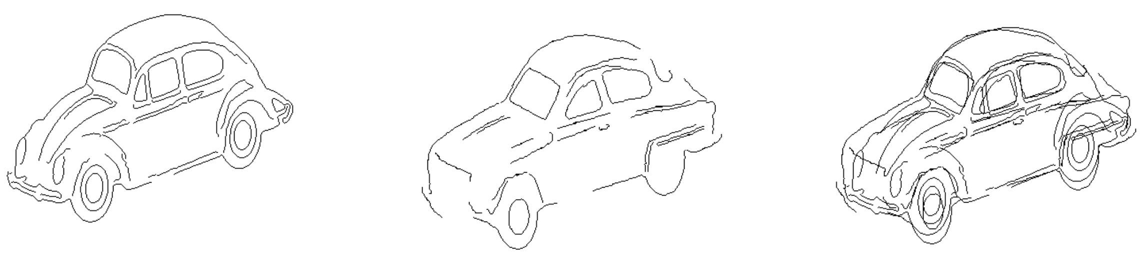

Figure 6: Matching a prototype car (Saab) to an image of a VW car.

Figure 7: Matching a prototype car (VW) to an image of a Saab car.

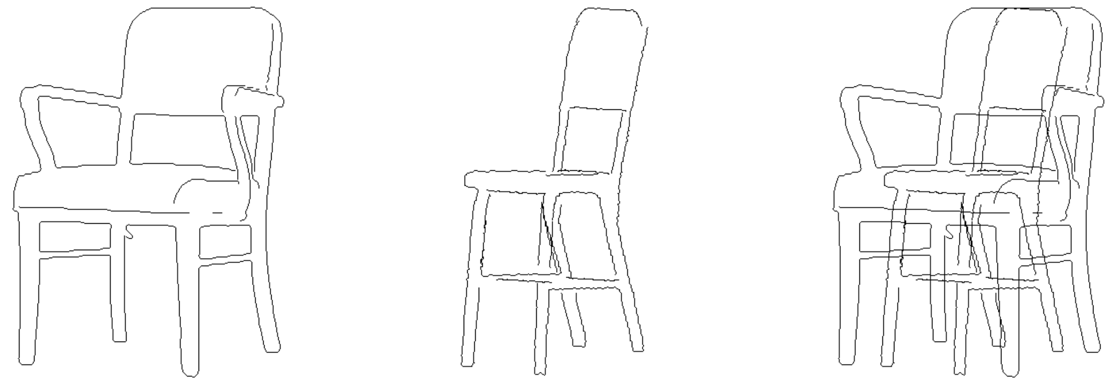

Figure 8: Matching a prototype chair (chair B) to an image of chair A with wrong correspondence.

Figure 9: Matching a prototype car (Saab) to an image of a VW car with wrong correspondence.

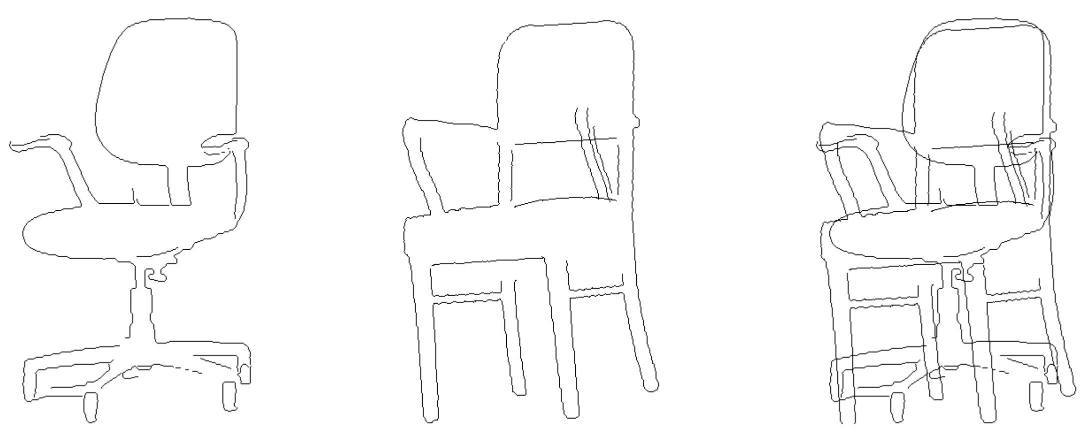

Figure 10: Matching a four-legged chair to an image of an office chair.

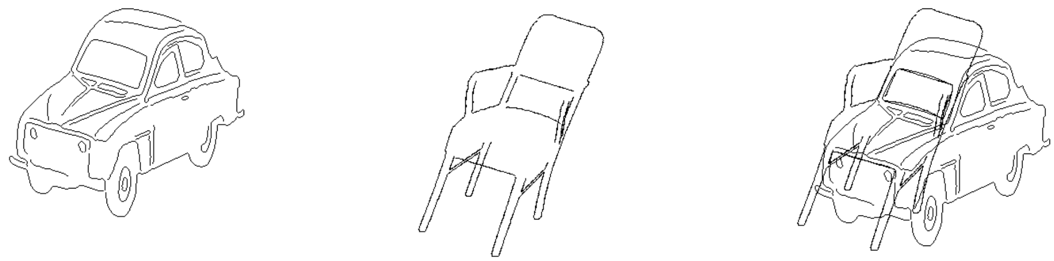

Figure 11: Matching a prototype to a chair (chair A) to an image of a Saab car.

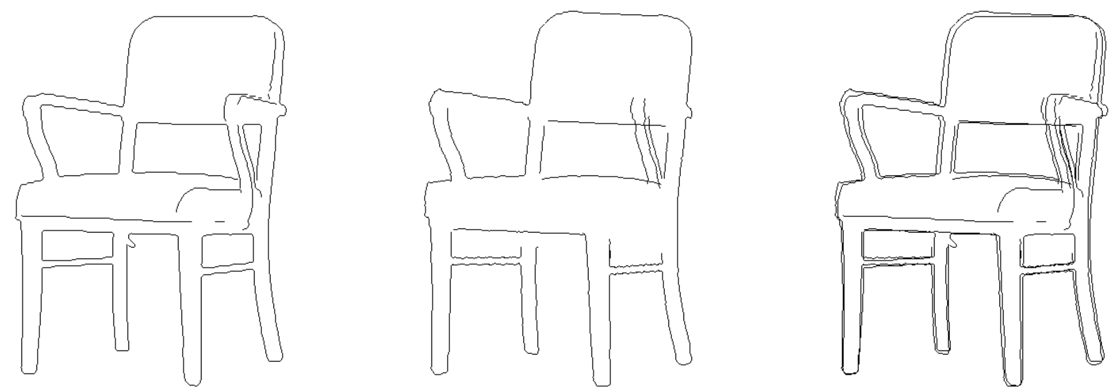

Figure 12: Matching a model of chair A to an image of the same chair using the prototype transform computed in the categorization stage.

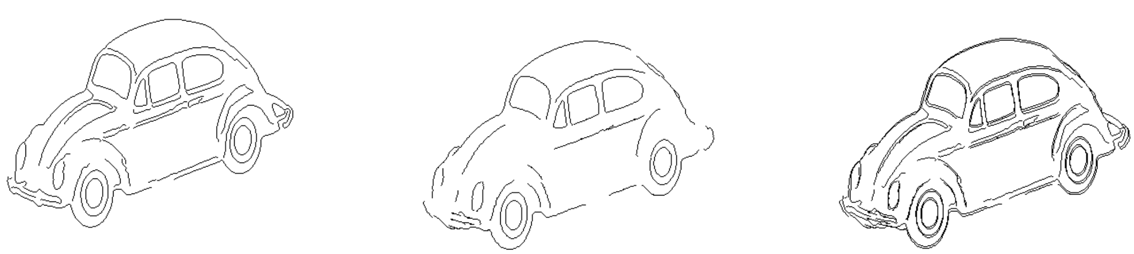

Figure 13: Matching a model of a VW car to an image of the same car using the prototype transform computed in the categorization stage.

presented an algorithm for constructing the optimal prototypes and discussed the relevance of the scheme to human recognition.

An important issue conveyed by our scheme is that categorization can be used to facilitate the identification of objects. We showed that by first categorizing the object, the difficult stages of the alignment process, namely, the recovery of the object pose and the correspondence between the image and the model, can be performed only once per class. Consequently, identification is reduced in this scheme into a series of simple template comparisons.

The scheme presented in this paper differs from existing categorization schemes in two important aspects. The existing schemes (e.g., [4]) first attempt to recover the part structure (geons) of the object from the image alone. This structure is assumed to be almost invariant both to rotation of the object and across objects of the same class. In contrast, our scheme does not attempt to recover any $3D$ information from the image alone. Moreover, it separates the two effects that determine the object's appearance: view variation effects and deformations due to class variability. View variations are compensated for by recovering the view of the prototype that most resembles the image, and the amount of deformation that separates the prototype from the specific object is evaluated by assessing the difference (in $2D$) between the nearest prototype view and the image.

Open problems for future research include solving the correspondence between prototypes and images, combining the scheme with existing indexing approaches, defining effective measures to evaluate the quality of matches, and extending the system to incorporate additional cues, such as color and texture.

Acknowledgement

I wish to thank Shimon Ullman for encouragement and advice, Tao Alter and Yael Moses for many fruitful discussions, Dror Bar Natan for his assistance in verifying the proof for Theorem 4, Eric Grimson, John Harris, and Tomaso Poggio for comments on earlier drafts.

References

[1] Anglin, J., 1976. Les premiers termes de référence de lénfant. In *S. Enrlich and E. Tulving (Eds.), La memoire sémantique*. Paris: Bulletin de Psychologie.

[2] Bajcsy R. and Solina F., 1987. Three dimensional object representation revisited. *Proc. of 1st ICCV Conference, London*, 231-240.

[3] Basri R. and Ullman S., 1988. The alignment of objects with smooth surfaces. *Proc. of 2nd Int. Conf. of Computer Vision, Florida*, 482-488.

[4] Biederman, I. 1985. Human image understanding: recent research and a theory. *Computer Vision, Graphics, and Image Processing, 32*, 29-73.

[5] Biederman, I., 1988. Aspects and extensions of a theory of human image understanding. In *Z. Pylyshyn (Ed.), Computational Processes in Human Vision: an Interdisciplinary Perspective*, Norwood, NJ: Ablexj, 370-428.

[6] Binford, T.O., 1971. Visual perception by computer. *IEEE Conf. on Systems and Control*.

[7] Brooks, R., 1981. Symbolic reasoning among 3-dimensional models and 2-dimensional images. *Artificial Intelligence, 17*, 285-349.

[8] Bruce, V., 1990. Perceiving and recognizing faces. *Mind and Language, 5(4)*, 342-364.

[9] Chien, C.H. and Aggarwal, J.K., 1987. Shape recognition from single silhouette. *Proc. of ICCV Conf., London*, 481-490.

[10] Davis L.S., 1979. Shape matching using relaxation techniques. *IEEE Trans. on Pattern Analysis and Machine Intel., 1(1)*, 60-72.

[11] Duda, R.O. and Hart P.E., 1973. Pattern classification and scene analysis. *Wiley-Interscience Publication, John Wiley and Sons, Inc.*

[12] Faugeras, O.D. and Hebert, M., 1986. The representation, recognition and location of 3D objects. *Int. J. Robotics Research 5(3)*, 27-52.

[13] Fischler, M.A. and Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Com. of the A.C.M., 24(6)*, 381-395.

[14] Forsyth, D., Mundy, J.L., Zisserman, A., Coelho, C., Heller, A., and Rothwell, C., 1991. Invariant descriptors for 3-D object recognition and pose. *IEEE Trans. on Pattern Analysis and Machine Intel., 13*, 971-991.

[15] Grimson W.E.L. and Lozano-Pérez T., 1984. Model-based recognition and localization from sparse data. *Int. J. of Robotics Research, 3*, 3-35.

[16] Ho, S., 1987. Representing and using functional definitions for visual recognition. *Ph.D. Dissertation, University of Wisconsin, Madison.*

[17] Hoffman, D. and Richards, W., 1984. Parts of recognition. *Cognition,*, 1865-1896.

[18] Huttenlocher, D.P., and Ullman, S., 1990 Recognizing Solid Objects by Alignment with an Image, *Int. J. Computer Vision, 5(2)*, 195-212.

[19] Humphreys G.W. and Riddoch M.J., 1987. To see but not to see. A case study of visual agnosia. *Laurence Erlbaum Associates, Pub., London.*

[20] Jacobs, D.W., 1992. Space efficient 3D model indexing. *Proc. of Image Understanding Workshop*, 717-725.

[21] Jolicoeur P., 1985. The time to name disoriented natural objects. *Memory and Cognition, 13(4)*, 289-303.

[22] Koenderink, J.J. and Van Doorn, A.J., 1982. The shape of smooth objects and the way contours end. *Perception, 11*, 129-137.

[23] Lamdan, Y., Schwartz, J.T., and Wolfson, H., 1987. On recognition of 3-D objects from 2-D images. *Courant Inst. of Math. Sci., Rob. TR 122.*

[24] Lissauer H, 1890. Fall von Seelenblindheit nebst einembeitrag zur theorie derselben. *Archives Psychiatrie und Nervenkrankheiten, 21*, 222.

[25] Lowe, D.G, 1985. Three-dimensional object recognition from single two-dimensional images. *Courant Inst. of Math. Sci., Rob. TR 202*

[26] Marr D and Nishihara, H.K, 1978. Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. of the Royal Society, London, B200*, 269-294.

[27] Nickerson, R.S., 1965. Short term memory for meaningful configurations: a demonstration of capacity. *Canadian J. of Psychology, 19*, 155-160.

[28] Palmer, S.E, Rosch, E, and Chase, P., 1981. Canonical perspective and the perception of objects. *In Long J. and Baddeley A. (Eds), Attention and Performance, 9, Erlbaum Hillsdale, NJ*, 135-151.

[29] Perret, D.I., Smith, P.A.J., Potter, D.D, Mistlin, A.J., Head, A.S., Milner, A.D and Jeeves, M.A, 1985. Visual cells in the temporal cortex sensitive to face view and gaze direction. *Proc. of the Royal Society, B223*, 293-317.

[30] Poggio, T., 1990. 3D object recognition: on a result by Basri and Ullman, *TR 9005-03, IRST, Povo, Italy.*

[31] Poggio, T. and Vetter T., 1992. Recognition and structure from one 2D model view: observations on prototypes, object classes, and symmetries. *M.I.T., A.I. Memo No. 1347.*

[32] Rosch, E, 1973. On the internal structure of perceptual and semantic categories In *T.E. Moore (Ed.), Cognitive development and the Acquisition of Language*, New York: Academic Press.

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

[34] Rosch, E, Simpson, C, and Miller, R.S., 1976. Structural bases of typicality effects. *J. of Experimental Psychology: Human Perception and Performance, 2(4)*, 491-502.

[35] Rosenfeld A, Hummel R, and Zucker S., 1976. Scene labeling by relaxation operations. *IEEE Trans. on System and Man Cybernetics, 7*, 420-433.

[36] Shapira, Y. and Ullman, S., 1991. A pictorial approach to object classification. *Proc. of the Int. Conf. on artificial intel.*, 1257-1263.

[37] Shepard, R.N, 1967. Recognition memory for words, sentences, and pictures. *J. of Verbal Learning and Verbal Behavior, 6*, 156-163.

[38] Standing, L, Conezio, J., and Haber, R.N, 1970. Perception and memory for pictures: single-trial learning stimuli. *Psychonomic Science, 19*, 73-74.

[39] Stark, L., and Bowyer, K, 1990. Achieving generalized object recognition through reasoning about association of function to structure. *IEEE Trans. on PAM, 13(10)*, 992-1006.

[40] Thompson, D.W and Mundy J.L., 1987. Three dimensional model matching from an unconstrained viewpoint. *Proc. of IEEE Int. Conf. on robotics and Automation*, 208-220.

[41] Ullman S., 1989. Aligning pictorial descriptions: an approach to object recognition. *Cognition, 32(3)*, 193-254.

[42] Ullman, S. and Basri, R, 1991. Recognition by linear combinations of models. *IEEE Trans. on PAM, 13(10)*, 992-1006.

[43] Vaina, L.M and Zlateva, S.D, 1990. The largest convex patches: a boundary-based method for obtaining object parts. *Biological Cybernetics, 62*, 225-236.

[44] Warrington E.K and Taylor A.M, 1973. The contribution of the right perietal lobe to object recognition. *Cortex, 9*, 152-164.

[45] Warrington E.K and Taylor A.M, 1978. Two categorical stages of object recognition. *Perception, 7*, 695-705.

[46] Weinshall, D, 1991. Model-based invariants for 3D vision. *Research Report RC-17358 (#76640), IBM T.J. Watson Research Center.*

[47] Winston, P.H, Binford, T.O, Katz, B, and Lowry, M, 1984. Learning physical description from functional definitions, examples and precedents. *M.I.T., A.I. Memo 679.*