Clustering Appearances of 3D Objects

Ronen Basri*
Dept. of Applied Math.
The Weizmann Inst. of Science
Rehovot, 76100, Israel

Dan Roth
Dept. of Computer Science
University of Illinois
Urbana, IL 61801

David Jacobs
NEC Research Institute
4 Independence Way
Princeton, NJ 08540

Abstract

We introduce a method for unsupervised clustering of images of 3D objects. Our method examines the space of all images and partitions the images into sets that form smooth and parallel surfaces in this space. It further uses sequences of images to obtain more reliable clustering. Finally, since our method relies on a non-Euclidean similarity measure we introduce algebraic techniques for estimating local properties of these surfaces without first embedding the images in a Euclidean space. We demonstrate our method by applying it to a large database of images.

1 Introduction

Perceptual categorization is one of the most intriguing problems in computer vision. One of the fundamental questions in categorization is what process can cause natural classes of objects to emerge from a set of unlabeled images. In an attempt to provide an answer to this question we introduce below a system that begins with a large number of unlabeled images (or sequences of images) of 3D objects and attempts to cluster the images according to the shape of the objects. Clustering images is important if we wish to automatically construct models of both classes and individual objects. In addition, it may provide insight into the way object categorization is implemented in the human visual system.

When we try to cluster objects by comparing their appearances we must take into account two problems. First, when we compare two images of two similar objects we may find that the images are very different from each other because the two images are taken under very different viewing conditions. Likewise, when we compare two images of two different objects we may find that due to the loss of information with projection the images are very similar to one another. Consequently, it is often difficult to determine whether the similarity measured between pairs of images indicates similar relationships between the objects, or whether it is merely an artifact of viewing conditions.

One possible way to circumvent this problem is by comparing a large set of images. When we compare many pairs of images of objects we may expect that the similarities between the objects will be reflected in the relationships between their sets of images. Our task, therefore, is to find effective ways to infer the similarities between objects from the collective similarities between the images.

In this paper we develop a system for clustering unlabeled images of objects according to their shape. Our method is based on the observations that objects produce images that in the space of all possible images form surfaces that are generally low dimensional and smooth. In addition, the surfaces of images produced by similar objects are often fairly close and parallel. We thus approach the problem of image clustering by introducing a general method for surface clustering whose objectives are to detect smooth surfaces and group together near-parallel surfaces. We further use sequences of images (*tracks*) to overcome non-smooth transitions in these surfaces and to resolve accidental intersections. The method we introduce can deal with similarity measures that are only locally Euclidean. In particular, we develop techniques for clustering that do not require embedding the images in a Euclidean space, but work directly with similarities.

We test the validity of our assumptions experimentally by applying the system to a fairly large database of images of 18 segmented objects. For the experiments we define a simple similarity measure, one that is based on measuring the distortion of local features. Our experiments demonstrate that using our method natural classes of objects emerge with high accuracy. These results indicate that surface clustering is a powerful mechanism that can be used to find useful clusters of images even when a simple similarity measure is used.

The paper contains the following sections. In Section 2 we briefly review the existing approaches to categorization. Section 3 lays out the principles of our clustering algorithm. Section 4 describes our algorithm in detail, and Section 5 offers experimental results.

2 Background

Most existing approaches to the categorization of 3D objects from 2D images look in the images for properties of the objects that are invariant over a wide range of viewing conditions. These include methods that extract global features of the objects and cluster the images according to these features [16, 7]. A

*This research was supported in part by a grant from the Israel Science Foundation No. 148/96. The vision group at the Weizmann Inst. is supported in part by the Israeli Ministry of Science, Grant No. 8504. Ronen Basri is an incumbent of Arye Dissentshik Career Development Chair at the Weizmann Institute.

second family of methods relies on the part structure of objects for categorization [5, 4, 18, 6, 20, 27, 10]. The underlying assumption of these methods is that objects that belong to the same perceptual category maintain roughly the same set of parts. Finally, there are methods which seek to interpret the perceived shapes in terms of their function [33, 13, 22, 28].

Unfortunately, it has proven difficult to extract invariant properties from images. Representations that rely on global properties tend to be sparse, and so they often are applied to problems that involve very few classes. Part structure efficiently characterizes many classes of interest. Nevertheless, many shapes are difficult to describe by parts (e.g., shoes). Also, part extraction from images tends to be sensitive to small changes of the shape, and many objects appear to produce different sets of parts from different aspects. Methods that rely on function suffer from similar problems. For this reason most existing studies of functionality were applied to 3D representations of objects rather than to their 2D projections. It is interesting to note that there is also an ongoing debate in the psychology literature as to whether perceptual categories are characterized by invariant properties (see [17]).

The difficulty in using invariance leads us to seek other mechanisms for categorization. The assumption underlying invariance-based approaches, that the properties which are essential for determining the class of objects can be detected in single images, is replaced by a method which determines the class of objects from large ensembles of images. Because a large number of images are considered it will be possible to obtain useful clusters even with a fairly simple similarity measure. Our motivation in using large data sets of images is driven in part by the progress in technology which makes the storage and comparison of large numbers of images feasible. In addition, images in large numbers are clearly available to the human visual system. The extent to which this large volume of images plays a role in perceptual categorization has not yet been determined.

Our solution to the problem of image clustering is based on detecting the smooth and parallel surfaces in the space of all images. Representing the images of objects as surfaces in a high dimensional space was the idea underlying several studies of recognition which attempt to identify individual instances of objects [9, 14, 19, 21, 29]. Similar ideas also appeared in studies which attempt to categorize objects using an a-priori known model or in the context of supervised learning (e.g., [31, 30, 8, 3]). These studies use supervision to derive a feature space in which the images of similar objects produce tight clusters. Unlike these studies, we address the problem of unsupervised clustering. Also unique to our method is the use of a non-Euclidean similarity measure (see [2, 15] for further insights to this problem). Finally, the idea of detecting smooth surfaces of images from a collection of single images and tracks is inspired in part by methods for curve extraction and perceptual grouping (e.g., [11, 23, 32, 34]). Our problem, however, is more difficult since we attempt to detect surfaces of arbitrary dimension in a high dimensional, non-Euclidean space.

3 Clustering Appearances

In this section we describe our solution to the problem of image clustering. We begin by explaining why image clustering can be recast as a problem of surface clustering. We next outline the steps of our algorithm and then describe these steps in detail.

In our method we assume that a large number of images are available to the system. When we consider a large number of images it is useful to think of the images of an object as a surface in the space of all possible images. Every image of the object will be a point on this surface. The dimension of the surface will generally be much lower than the dimension of the space [9, 14, 19, 21, 24, 29], but it may be arbitrary, due to changes in lighting, viewpoint, articulation, etc., and may even vary at different places. (In fact, the set of images may even have volume in space due, e.g., to lighting variations, see [1]). In addition, the surface may self intersect, e.g., due to symmetries of the object. In general, we may expect the surfaces produced by the set of images of objects to typically be continuous and slowly curving. The surfaces will be continuous since small changes in the viewing parameters will generally produce only small changes in the appearance of the objects. This will be generally true except at the boundaries of very different aspects of an object, when a small rotation of the object may change its appearance drastically. The assumption that the surfaces are smooth amounts to the assumption that small changes in viewing condition have a roughly linear effect on the appearance of objects. This means, for example, that if moving the light source by a tiny amount changes the appearance of an object in a particular way (e.g., makes some patches darker and others brighter), then a further tiny motion of the light source will change the image at a similar rate. Although the assumption of smoothness is violated in some circumstances, we expect it to be true in general and use it as a working hypothesis, which we need to validate experimentally. This smoothness assumption is known to be exactly true of lighting and viewpoint changes for some limited circumstances ([29, 24]).

An important issue for clustering is the relation between surfaces representing the images of different objects. When two shapes are similar we may anticipate that all corresponding projections of these shapes seen under identical viewing conditions will also be similar. This implies that the two surfaces representing the images of these shapes will be relatively close to each other in most places. The actual distance between the surfaces may vary from place to place, but not by much. In contrast, when two shapes are very different we may expect that most of their projections will not be similar. As a result the surfaces representing their images will generally be distant from one another. An exception occurs when accidental (or nearly accidental) views exist, in which case the two surfaces may cross each other, or for a small section become close to one another.

To cluster the images of similar objects we need to detect the nearly parallel surfaces and distinguish

them from surfaces that accidentally cross one another. To perform this clustering we can use the following procedure. First, we identify local patches on the surfaces and estimate their dimension and orientation. Then, we attempt to determine what set of surface patches represent the same individual object. Patches of low dimension will tend to correspond to views of a single object, whereas patches of high dimension may indicate the presence of an accidental intersection of surfaces representing the images of different objects. In addition, pairs of patches that form smooth continuations are likely to come from a single object. Next, we attempt to connect surfaces that represent similar objects by identifying patches that are close to each other and have similar orientation. In our implementation we combine both smooth continuation and parallelism into a single affinity measure that reflects the evidence that two patches come from a single class.

The analogy to surface clustering demonstrates why standard pattern recognition approaches to clustering fail to cluster images. To illustrate this consider the images of two objects that share an accidental view. The trajectories of these objects in the space of all images near the location of their intersection form a cross-like shape. Standard clustering algorithms are not designed to separate the two lines of a cross.

An important source of information for image clustering is found in sequences of images. Tracks provide a reliable indication that their images are projections of the same individual objects. Thus, we may integrate the information which indicates the preferred clustering for all the images in a track to obtain a more reliable clustering solution. The use of tracks is particularly useful if their images lie near a non-smooth transition in the surface representing the object. In addition, tracks can resolve accidental regions of intersection of the sets of images of two different objects. The role of tracks in surface clustering resembles the role of curve fragments in perceptual grouping. Many subjective contours are easier to perceive when curve fragments are available (as opposed to only a sparse set of points), in particular when the available fragments include the corners and high curvature sections of the boundaries of shapes [4].

Based on these observations we propose the following algorithm for image clustering. Given a set of images or sequences of images we first compute the similarities between all pairs of input images. Next, for every image we select the images that are most similar and use them to estimate the local orientation and dimension of the surface unit that includes the image. We then consider every pair of surface units and compute an affinity measure that reflects the distance between the units and their relative orientation. Subsequently, for every pair of tracks we compute an affinity measure by integrating the affinities between their surface units. Finally, we turn our problem into a graph partitioning problem by applying a standard clustering algorithm to a graph obtained by assigning weights according to the affinities between the tracks.

In the next section we assume that the similarities between the images are already given and proceed to formalize the steps of the clustering algorithm. The similarities are assumed to locally be Euclidean and roughly linear. We will verify the accuracy of this assumption for a particular similarity function in Sec. 5. Based on these assumptions we describe a method for estimating the dimension and orientation of surface units directly from the similarities without embedding them first in a Euclidean space. We then use these estimates to assign affinities between tracks and perform the clustering.

4 Computing affinities between tracks

In this section we describe how to compute the affinities between tracks based on the similarities between the images. Since it is desired that the affinities between tracks will reflect the distance and relative orientation between their surface units we will need to describe how these can be estimated. The difficulty is that the similarities between images are not Euclidean and therefore it may not be possible to embed the images in a Euclidean space without distorting the similarity values. A common method to overcome this problem is to use multidimensional scaling (MDS) to first embed the images in a Euclidean space in a way that minimizes the necessary distortion of the distances [25]. MDS, however, is an iterative optimization process that often converges to a local minimum, and so it may be slow and unreliable. As an alternative, we show below how we can estimate the dimension and orientation of surfaces directly from the distances without first embedding them in space.

4.1 Estimating dimension

We assume that the similarities between the images are expressed as distances, that is, they are non-negative and vanish for identical images. Given such distances we turn to estimating the dimension of surface units. The term *surface unit* is used here to denote a surface patch around a given image, which we estimate from the set of nearby images. We next show how the dimension of surface units can be estimated directly from the distances.

Let $p_1, ..., p_n$ be n points in $\mathcal{R}^d$ and let p_0 denote the origin. Suppose we wish to determine the surface that passes through p_0 whose distance to $p_1, ..., p_n$ is minimal. Denote by P a $d \times n$ matrix whose columns are $p_1, ..., p_n$. Then the dimension of the surface can be found by looking at the eigenvectors and eigenvalues of the *scatter matrix* PP^T, where the dominant eigenvectors point to the principal orientations of the surface and the other eigenvectors point to directions in which the surface is thick or curved.

Another matrix that is related to the scatter matrix is the *Grammian matrix*, P^TP. The Grammian matrix has exactly the same eigenvalues as the scatter matrix, and their corresponding eigenvectors are related by P, since $P^TPx = \lambda x$ implies $PP^TPx = \lambda Px$. Consequently, if x is an eigenvector of the Grammian matrix with an eigenvalue λ then Px is an eigenvector of the scatter matrix with the same eigenvalue. The Grammian matrix contains the inner products between all the pairs of points $p_1, ...p_n$. These inner products can be recovered from the distances between

triplets of points. Given three points o, u, and v, let o denote the origin, the inner product between $u = u - o$ and $v = v - o$ can be computed as follows:

$$\|v - u\|^2 = \|u\|^2 + \|v\|^2 - 2u^T v.$$

Therefore,

$$u^T v = \frac{1}{2}(\|u\|^2 + \|v\|^2 - \|v - u\|^2),$$

and consequently

$$u^T v = \frac{1}{2}(d_{uo}^2 + d_{vo}^2 - d_{uv}^2),$$

where the notation d_{uv} represents the distance between the points u and v. Notice that this way each component of the Grammian matrix is determined by a small number of points (up to three points). Therefore, if only a few of the distances are corrupted they will affect only a small portion of the Grammian matrix.

The process of building the Grammian matrix requires us to choose an origin. In general, we want to take the centroid of the points to be the origin. Denote by $\hat{P}$ the matrix P after its columns are translated to bring their centroid to the origin. It can be readily verified that $\hat{P} = PC$ where $C = I - \frac{1}{n}11^T$, and $1 \in \mathcal{R}^n$ is a vector whose components are all 1's. Thus we need to multiply the Grammian matrix by C from both sides.

Once the eigenvalues of the Grammian matrix are recovered the dimension of the underlying surface unit can be estimated. In the experiments below we allow our objects to rotate in two directions. We thus expect the surface units to be two-dimensional. If we find the dimension of a unit to be higher than two it may indicate that the images in this unit come from more than a single object. We can thus rank the surface units by the ratio between the second largest and third largest eigenvalues. The larger this measure is, the more likely it is that the surface is two-dimensional.

4.2 Estimating relative orientation

Next, we want to determine the relative orientation of two surface units. Given two linear subspaces the angles between them can be estimated as follows. Let A and B be two $d \times n$ and $d \times m$ matrices whose columns are orthonormal and span the two spaces. The cosines of the angles between the two surfaces are given by the singular values of $B^T A$ (see, e.g., [12], pp. 584–585). Denote the points which determine the two surfaces by $p_1, ..., p_n$ (with the origin set at p_0) and by $q_1, ..., q_m$ (with the origin set at q_0), and denote their associated matrices by P and Q respectively. In our case we face two problems since P and Q are unknown and since their columns are not orthonormal. Nevertheless, we can recover the angles as follows.

A and B contain orthonormal representations of the two surfaces. Such representations may include the dominant eigenvectors of the scatter matrices associated with the surfaces, PP^T and QQ^T respectively. Recall that these eigenvectors are related through P (and Q) to the corresponding eigenvectors of the Grammian matrix. Thus, the columns of PX (where X is a matrix whose columns contain the dominant eigenvectors of $P^T P$) provide an orthogonal (but not necessarily orthonormal) basis to the surface. To normalize this basis we need to divide each column by $\|Px\| = \sqrt{\lambda}$. Let $D_\lambda = \text{diag}\left(1/\sqrt{\lambda_1}, ..., 1/\sqrt{\lambda_n}\right)$, where $\lambda_1, ..., \lambda_n$ are the eigenvalues of $P^T P$, then we may write $A = PXD_\lambda$. Similarly, we may write $B = QYD_\mu$, where Y is a matrix whose columns contain the dominant eigenvectors of $Q^T Q$, $D_\mu = \text{diag}\left(1/\sqrt{\mu_1}, ..., 1/\sqrt{\mu_m}\right)$, and μ_i are the eigenvalues of $Q^T Q$. Thus,

$$B^T A = D_\mu Y^T Q^T P X D_\lambda.$$

The eigenvectors and eigenvalues of the two scatter matrices are known at this stage, so what is left to recover is the matrix $Q^T P$. This matrix contains inner products of the form $(q_j - q_0)^T (p_i - p_0)$. These inner products can be recovered from distances between quadruples of points, as follows. Given four points, a, b, u, and v the inner product between $u - a$ and $v - b$ is given by

$$(u - a)^T (v - b) = \frac{1}{2}(d_{ub}^2 + d_{va}^2 - d_{uv}^2 - d_{ab}^2).$$

Finally, in this case too we need to choose an origin. Again, we set the origin at the centroid of the points by multiplying $Q^T P$ by C from both sides.

4.3 Computing affinities and clustering

Based on the dimension and relative orientation of surface units we build the affinities between surface units as follows. Let $r(u)$ denote the score assigned to a unit u reflecting its dimension. Let d_{uv} denote the distance between the units (we take this to be the distance between the two images around which the units were formed), and let $\alpha_1, ..., \alpha_n$ denote the angles between the units (in our experiments $n = 2$) then we define:

$$C(u, v) = e^{d_{uv}^2/\sigma - \alpha_1^2/\rho_1 - ... - \alpha_n^2/\rho_n},$$

for some constants $\sigma, \rho_1, ..., \rho_n$. The affinity between u and v is defined as

$$A(u, v) = C(u, v) r(u) r(v).$$

To obtain the affinities between two tracks we sum $A(u, v)$ over all pairs of units in the two tracks.

Once we obtain the affinities between the tracks we build a complete graph whose nodes represent the tracks to be clustered and set the weights of the edges to be the affinities between the tracks. At this point we treat the problem as a standard graph clustering problem. In our experiments we used a recursive application of a normalized cut algorithm (as used in [26]) to partition the graph. This produces a binary tree in which the hierarchy of the clustering is reflected in the levels of the tree.

5 Experiments

In this section we describe the experiments conducted to validate our method. We begin by briefly describing the similarity measure used, which penalizes for the distortion of local features. As we demonstrate in our experiments the measure is strongly affected by viewing conditions and deteriorates fairly quickly with a change in viewing position. Consequently, we will show that standard clustering algorithms when given this measure fail to detect satisfactory clusters of images. The measure, however, is fairly smooth, and so we can use it to produce affinities between tracks in the manner described in the previous section. We will show that using our method, when applied to a database of 1710 segmented images of 18 objects, natural classes of objects emerge. Finally, we will show that already with tracks of moderate lengths we manage to achieve excellent classification results.

5.1 Similarity between images

Our measure of similarity is based on measuring the distortion of salient local features between images. While we restrict the scope of this paper to segmented images, we have chosen a similarity measure that relies on local features in the expectation that it can be extended in the future to deal with segmentation errors and occlusion.

Formally, we identify salient features using a window of 16×16 pixels. For every such window in the image we measure the variance of grey-level values and select those windows which have maximal variance. To reduce the amount of computation whenever two selected windows are very close to each other (less than four pixels away) we keep only the one with higher variance. Once we have selected the salient windows we normalize their grey level values by bringing their means to zero and variance to one. Then, for every selected window in one image we compare it to *all* windows (not only the salient ones) in proximate locations in the other image. Given two windows let d denote the distance between their location, and let $r_1, ..., r_4$ denote the Euclidean difference between their normalized grey-values at four different scales, then we define the similarity between the two windows, w_1 and w_2, as

$$S(w_1, w_2) = e^{-(d^2/\sigma + r_1^2/\rho_1 + ... r_4^2/\rho_4)}$$

with $\sigma = 1250$ and $\rho_1 = ... = \rho_4 = 1$. Then, for every salient window in one image we maximize this functional over all windows in the other image, yielding:

$$S(w_1) = \max_{w_2} S(w_1, w_2).$$

Finally, we define the similarity between the two images, $S(I_1, I_2)$, to be the average of all $S(w)$ taken over all salient windows in both images.

The similarities defined above always return values between zero and one. They return one when applied to two identical images. When we rotate an object slightly the similarity between the images degrades until it reaches the level of noise. This produces a bell shaped function (see Fig. 1(left)). This is a typical behavior of so called *quasi-invariant* measures, where the width of the bell indicates the speed of degradation of the chosen measure.

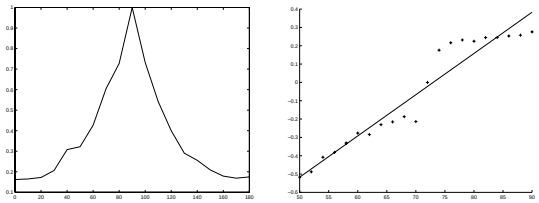

Figure 1: Left: The similarities between a side view of a shoe (90°) and other images of the same shoe obtained by horizontal rotations. Right: linear regression of the similarity values between images of a CAD model of a cow taken under rotation of $\pm 20°$ in multiples of 2°.

After computing the similarities we would like to convert them to distances. The distance between any two images should be non-negative, and vanish for identical images. We achieve this by defining: $D(I_1, I_2) = -\log S(I_1, I_2))$.

Our method assumes that the distance measure is roughly linear locally. An example of a linear regression for an object rotated by small amounts is shown in Fig. 1(right). Notice that our distance measure is non-Euclidean and does not even form a metric. The process of evaluating the distance between two images involves for every salient feature a search for the best corresponding feature in the other image. This process is not guaranteed to find a corresponding feature or to keep consistent correspondences in different comparisons. Thus, it is not difficult to produce examples that violate the triangular inequality.

5.2 Results

To test our method we have collected images of 18 objects (Fig. 2). For every object we took 95 images according to the following procedure. The objects were put on a turntable that was rotated about the vertical axis by multiples of 10° from 0° to 180° providing 19 images per object. A camera mounted on a robotic arm was rotated around the horizontal axis of the object to five positions each differing by 10°. The total number of images in our database, therefore, was $1710 = 18 \times 19 \times 5$. The objects were put before a turquoise background cloth to allow their complete automatic segmentation. After segmentation the images were translated and scaled uniformly so that the object would fit a square of 250×250 pixels. The images were then converted to black-and-white, and the background intensity was set to three standard deviations below the mean of the grey level values of the object. We then compared all pairs of images to determine the similarities between them.

Below we examine our results with respect to five classes that emerged from the experiments, shoes, cars (including the truck), vegetables, wild cats, and thick-skinned animals (hippopotamus and rhinoceros). Success rates were evaluated with two common measures, *accuracy* and *purity*. Given the images of a certain class and given a computed cluster, accuracy is the fraction of class members that are included in the cluster. Purity is the fraction of clustered images that be-

l/n	8/12	6/16	4/24	2/48	1/95
Omit	4	8	17	29	23
Shoes	100(100)	98(100)	93(100)	83(89)	62(100)
Cars	100(98)	100(97)	100(96)	96(94)	87(90)
Veg.	100(100)	98(100)	100(100)	99(99)	100(98)
Cats	98(100)	97(100)	98(100)	94(100)	86(100)
Thick	100(100)	98(100)	87(100)	81(100)	78(100)
Mean	99(100)	98(100)	96(99)	91(96)	81(98)
+kNN	99(100)	98(99)	95(98)	86(90)	74(96)

Table 1: Applying our method to the single images (right column) and to random tracks from the database (averages over 20 runs). Top row: mean length and number of tracks. Second row: tracks reported still unclassified in the first, clustering stage of our algorithm. Bottom row: performance after these tracks are classified using k-nearest neighbors.

l/n	8/12	6/16	4/14	2/48	1/95
Mean	93(85)	88(81)	85(75)	76(71)	68(68)

Table 2: Mean performance of our method when tested against the images of single objects.

long to the class. High accuracy indicates that most images of that class were clustered together, while high purity indicates a small number of false positives. We measure accuracy and purity for every class by selecting the cluster that maximizes the product of these two measures.

Table 1 shows the result of applying our method to the database. In typical applications the five classes emerged as the top-most clusters. Already when the method was applied to single images a significant improvement over the standard algorithm was obtained. The high purity values, in particular, indicate that there was a tendency to split classes rather than to confuse between classes. When the method was applied to tracks of moderate lengths a near perfect clustering was obtained.

One difficulty in evaluating our results stems from the following problem. In our method we estimate the dimension and orientation of surface units. To avoid instabilities in this process we insisted on having sufficiently many images in each neighborhood. This led to throwing away a significant number of images from the database. To control for this problem we classified the omitted tracks using a k-nearest neighbors algorithm. As can be seen in Table 1, with tracks of moderate lengths there was no noticeable difference in the performance.

Finally, Table 2 shows the result of detecting the images of individual objects with our method. In contrast with the classification results we see here that many objects were confused with other objects of the same class. This is far from surprising. In an analogy to perceptual grouping consider an image containing sets of parallel curve fragments. Any attempt to complete such fragments to curves will necessarily be problematic because every fragment will find several almost equally good completions. The same happens with our clustering algorithm.

6 Conclusion

We have addressed the problem of clustering unlabeled images of 3D objects in an attempt to develop a method that will cause natural classes of objects to emerge. Unlike existing approaches, our method does not rely on extracting properties of the objects that are invariant to changes in viewing conditions. Instead, we have argued that the problem of clustering images can be solved by considering a large number of images of objects provided that the method of clustering properly accounts for the relationships between the sets of images of the objects. Our method is based on the observation that image clustering resembles the problem of perceptual grouping of points and curve fragments in images. Consequently, we have developed a method to partition the images into slowly curving and parallel surfaces. We further use tracks of images to overcome non-smooth transitions in these surfaces and to resolve accidental intersections. We have tested our algorithm on a fairly large database of segmented images and demonstrated that the method is capable of recovering natural classes of objects with very few false positives.

A significant portion of the paper was devoted to dealing with a non-Euclidean similarity measure. Many existing systems compute similarities using some ad-hoc algorithm that does not guarantee that the obtained similarities obey the metric rules. This in fact is the case also with our similarity measure. We circumvent this problem by assuming that the measure of similarity is roughly Euclidean locally, and by developing methods to estimate the dimension and orientation of the surfaces which represent the images of objects directly from the distances. Our experiments demonstrate the validity of this assumption.

Our clustering algorithm relies on a similarity measure that is based on measuring the distortion of local features. We chose to use this measure because we wanted a measure that could deal, in principle, with segmentation errors and partial occlusion, and we intend in the future to test it with such data. However, we acknowledge that local features fail to capture important information about shape, and we can foresee the use of other, more sophisticated measures, such as ones that consider the apparent part structure of the object (without assuming that part structure is invariant to viewing conditions), in a similar framework of clustering.

Examining the results of a clustering algorithm when applied to common shapes is not a straightforward task. When people examine such results they bring to mind all their past experience which leads them to categorize objects the way they do. This experience may rely on non-visual cues, color, texture, context, and other sources of information that extend beyond the scope of the tested algorithm. A further complication is that the quality of the clustering is not independent of the specific objects on which it was tested. The experiments demonstrate that our method is capable of detecting natural classes for a variety of objects. Nevertheless, we intend in the future to test the algorithm on larger data sets of images in order to obtain a better evaluation of its performance.

Figure 2: The objects (5 shoes, 2 cars, a truck, 2 peppers, 2 onions, a lion, a lioness, two tigers, a hippopotamus, and a rhinoceros). The objects are shown in different views to illustrate the variability of our database.

Finally, running the clustering algorithm on all 1710 images required significant computational resources, since it involved 1710×1710 comparisons of image pairs. This complexity is impractical if we wish to consider significantly more objects in the database or to accumulate larger numbers of images for each object (e.g., in order to deal with varying illumination conditions or non-rigidities). Nevertheless, our computations are essentially local, in the sense that only similarities between pairs of images that resemble each other matter for the computation. This implies that in principle we do not have to compute the similarities between all pairs of images, but to consider only potential candidates that may resemble one another. We intend in the future to study mechanisms to reduce the amount of computation required by the method.

References

[1] P.N. Belhumeur and D.J. Kriegman. 1996. What is the set of images of an object under all possible lighting conditions? *CVPR*:270–277.

[2] M. Brand, 1996. A fast greedy pairwise distance clustering algorithm and its use in discovering thematic structures in large data sets. *MIT Media Lab, Tech Rep 406.*

[3] C Bregler and S. Omohundro, 1995. Nonlinear Manifold Learning for Visual Speech Recognition. *ICCV*:494–499.

[4] I. Biederman, 1985. Human image understanding: recent research and a theory. *CVGIP*, **32**:29–73.

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

[6] R. Brooks, 1981. Symbolic reasoning among 3-dimensional models and 2-dimensional images. *AI*, **17**:285–349.

[7] R.O. Duda and P.E. Hart, 1973. *Pattern classification and scene analysis.* Wiley and Sons, Inc.

[8] S. Edelman and S. Duvdevani-Bar, 1997. A model of visual recognition and categorization. *Phil. Trans. R. Soc. Lond. (B)*, **352**,(1358):1191–1202.

[9] O. Faugeras and L. Robert, 1996. What Can Two Images Tell Us about a Third One?. *IJCV*, **18**(1):5–19.

[10] M.M. Fleck, D.A. Forsyth, and C. Bregler, 1996. Finding naked people. *ECCV*:593–602.

[11] G. Guy and G. Medioni, 1996. Inferring Global Perceptual Contours from Local Features, *IJCV*, **20**(1/2):113–133.

[12] G.H. Golub and C.F. van Loan, 1989. *Matrix Computations.* The Johns Hopkins Univ. Press.

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

[14] D.W. Jacobs, 1997. "Matching 3D models to 2D images, *IJCV* **21**(1/2): 123–153.

[15] D.W. Jacobs, D. Weinshall, and Y. Gdalyahu, 1998. Condensing Image Databases when Retrieval is Based on Non-Metric Distances. *ICCV*: 596–601.

[16] A.K. Jain, 1988. *Algorithms for clustering data.* Prentice Hall.

[17] G. Lakoff, 1987. *Women, fire, and dangerous things.* Univ. of Chicago Press.

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

[19] H. Murase and S. Nayar, 1995. Visual learning and recognition of 3D objects from appearance. *IJCV*, **14**(1):5–25.

[20] Pentland, A., 1987, Recognition by Parts. *ICCV*:612–620.

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

[22] E. Rivlin, S. Dickenson, and A. Rosenfeld, 1994. Recognition by Functional Parts, *CVPR*:267–275.

[23] E. Sharon, A. Brandt, and R. Basri, 1997. Completion energies and scale. *CVPR*:884–890.

[24] A. Shashua, 1997. On photometric issues in 3D visual recognition from a single 2D image. *IJCV*, **21**(1/2):99–122.

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

[26] J. Shi and J. Malik, 1997. Normalized cuts and image segmentation. *CVPR*:731–737.

[27] K. Siddiqi and B.B. Kimia, 1995. Parts of visual form: computational aspects. *PAMI*, **17**(3):239–251.

[28] L. Stark and K. Bowyer, 1991. Achieving generalized object recognition through reasoning about association of function to structure. *PAMI*, **13(10)**:992–1006.

[29] S. Ullman and R. Basri, 1991. Recognition by linear combinations of models. *PAMI*, **13**(10):992–1006.

[30] T. Vetter, M.J. Jones, and T. Poggio, 1997. A bootstrapping algorithm for learning linear models of object classes. *CVPR*:40–46.

[31] T. Vetter and T. Poggio, 1997. Linear object classes and image synthesis from a single example image. *PAMI*, **19**(7):733–742.

[32] L.R. Williams and D.W. Jacobs, 1997. "Stochastic Completion Fields: A Neural Model of Illusory Contour Shape and Salience," *Neural Computation*, **9**: 837–858.

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

[34] S.W. Zucker, C. David, A. Dobbins, and L. Iverson, 1988. The Organization of Curve Detection: Coarse Tangent Fields and Fine spline Coverings. *ICCV*:568–577.