

Minimally Supervised Acquisition of 3D Recognition Models from Images *

Andrea Selinger and Randal C. Nelson
Department of Computer Science
University of Rochester
Rochester, NY 14627
{selinger,nelson}@cs.rochester.edu

Abstract

Image-based object recognition systems developed recently don't require the construction of a 3D geometric model, allowing recognition of objects for which current geometric recognition technologies do not apply. Such systems are typically trained with labeled, clean views that cover the whole viewing sphere and can sometimes handle generic, visually similar classes with moderate variation. It has been little explored whether such systems can be trained from imagery that is unlabeled, and whether they can be trained from imagery that is not trivially segmentable.

In this report we investigate how an object recognition system developed previously can be trained from clean images of objects with minimal supervision. After training this system on a single or a small number of views of each object, a simple learning algorithm is able to attract additional views to the object representation, building clusters of views belonging to the same object. We explore how the learning performance improves by extending the set of views, introducing a small amount of supervision, or using more complicated learning algorithms.

*Support for this work was provided by ONR grant N00014-96-1-0671, NSF grant CDA-9401142, and NSF grant IIS-9977206

1 Introduction

Recent results in object recognition research now make it possible to train 3D object recognition systems from imagery. This allows the recognition of objects without the requirement of constructing a 3D geometric model, and the recognition of objects for which current geometric recognition technologies do not apply. In various forms, image-based methods are currently the most successful general approach. Several of these systems can handle 5 or 6 continuous dimensions, usually orthographic freedoms, for individual objects. Some are also able to handle generic, visually similar, classes with moderate variation, for example airplanes, cars, or faces. Typically, a system is trained with a fully labeled (identity and pose) set of clean (segmentable) example views. The view set generally covers variation due to out-of plane rotation, and sometimes geometric distortion among class exemplars.

One of the first appearance based systems was the one developed by Poggio that recognized wire objects [10]. Rao and Ballard [11] describe an approach based on the memorization of the responses of a set of steerable filters. Mel’s SEEMORE system [6] uses a database of stored feature channels representing multiple low-level cues describing contour shape, color and texture. Schiele and Crowley [12] used histograms of the responses of a vector of local linear neighborhood operators. Murase and Nayar [7] find the major principal components of an image dataset, and use the projections of unknown images onto these as indices into a recognition memory. This approach was extended by Huang and Camps [4] to appearance-based parts and relationships among them. Wang and Ben-Arie [15] were able to do generic object detection using vectorial eigenspaces derived from a small set of model shapes which are affine transformed in a wide parameter range. The approach taken by Schmid and Mohr [13] is based on the combination of differential invariants computed at keypoints with a robust voting algorithm and semilocal constraints.

Two issues that have been little explored are (1) whether such systems can be trained from imagery that is unlabeled, and (2) whether they can be trained from imagery that is not trivially segmentable.

A recognition system that could be trained from either unlabeled or unsegmented imagery would be valuable for reducing the effort required to obtain a training set. Of greater practical impact, a 3D recognition system that could be trained from cluttered imagery would be useful for automatic, object-level labeling of image databases, which is an important outstanding problem.

In this report we take a first step towards training an object recognition system from unlabeled imagery. We use an object recognition system developed previously [9], and investigate how it can be trained from clean, black background images of objects with minimal supervision.

Our work is based on the fact that if the recognition system is trained on a few views of an object, it will be able to correctly recognize views that are topologically close to the original ones. After adding these views to the object representation the system will be able to recognize additional views in an iterative process. This will lead to the development of clusters of views characterizing each object. We look at the way these clusters develop and at how well they cover the view set. We investigate how the performance of such a learning system can be improved by strategies such as extending the set of initial views, increasing the sampling rate, introducing a small amount of user supervision, or by devising more

complicated learning algorithms.

2 The object recognition system

The recognition system that we use is based on a hierarchy of perceptual grouping processes [14]. A 3D object is a fourth-level group consisting of a topologically structured set of flexible 2D views, each derived from a training image. In these views, which represent third-level perceptual groups, the visual appearance of an object is represented as a geometrically consistent cluster of several overlapping local context regions. These local context regions represent high-information second-level perceptual groups, and are essentially windows centered on and normalized by key first-level features that contain a representation of all first-level features that intersect the window. The first level features are the result of first level grouping processes run on the image, typically representing connected contour fragments.

In more detail, distinctive local features called *keys*, selected from the first level groups in our hierarchy, seed and normalize *keyed context regions*, the second level groups. In the current system, the keys are contours automatically extracted from the image.

The second level of grouping into keyed context patches amplifies the power of the key features by providing a means of verifying whether the key is likely to be part of a particular object. This is necessary because the invariant parameters of the key features are relatively weak evidence. If only this weak evidence is used in an evidence combination scheme, a proliferation of high-scoring false object hypotheses results, especially when applied to complex, cluttered scenes, or when using features that are subject to partial occlusion (such as edges) [2]. The fact that the high-information units are overlapping fragments of a view gives robustness to clutter and occlusion not present in other image-based methods.

Even these high information local context regions are generally consistent with several object/pose hypotheses; hence we use the third-level grouping process to organize the context patches into globally consistent clusters that represent hypotheses about object identity and pose. This is done through a hypothesis database that maintains a probabilistic estimate of the likelihood of each third level group (cluster) based on statistics about the frequency of matching context patches in the primary database. The idea is similar to a multi-dimensional Hough transform without the space problems that arise in an explicit decomposition of the parameter space. In our case, since 3-D objects are represented by a set of views, the clusters represent two dimensional rigid transforms of specific views. As mentioned above, the use of keyed contexts rather than first-level groups gives the voting features sufficient power to substantially ameliorate well known problems with false positives in Hough-like voting schemes.

The system was trained on labeled imagery of a variety of 3D shapes, ranging from sports cars and fighter planes to snakes and lizards. These images were taken at about 20 degrees apart over the whole viewing sphere. When trained on 24 objects the system achieved a recognition performance of 97% on images that were taken in between the training views, under the same good conditions [8]. Performance remains relatively good in the case of clutter and partial occlusion [9].

Besides good recognition results, the system has properties that lend themselves to the

problem of training from unlabeled images. In particular it displays an ability to extrapolate to new poses of known objects, and an ability to generalize to similarly shaped, but previously unencountered objects, and robustness to clutter.

3 Training the system

In our baseline experiment we labeled 1 to 4 of our training images and used them as seeds in training. The recognition system was able to extract the neighboring views from a mixed database of images. We added these views to the existing object representation and we iterated the process.

3.1 The image corpus

We used an image corpus containing 583 clean, black background images of 6 objects (the cup, toy-bear, sports-car, toy-rabbit, plane and fighter plane seen in Figure 1) taken at about 20 degrees apart over the viewing sphere. We had 106 images per object (53 images per object hemisphere), except the sports-car that had only 53 images (we covered only the top hemisphere since the bottom hemisphere was flat and black, and thus uninteresting).

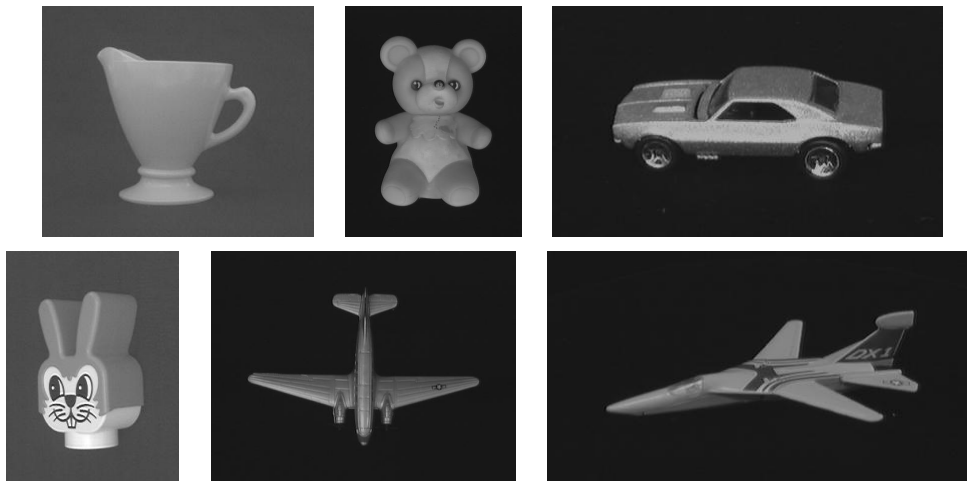


Figure 1: The objects used in testing the system.

In order to investigate the effect of sampling density we obtained an extended image set, with a denser sampling of the viewing sphere. This set has a total of 1035 clean, black background images taken at 15 degrees apart over the viewing sphere. There are 105 images of the sports-car and 186 images of each of the other objects. In this case the number of images of the sports car is not exactly half of the number of images for the other objects because we took images around the equator of the viewing sphere. Given that our recognition system is designed to handle a maximum rotation of 15 degrees from any training sample rather than the 20 degrees between the views in our original image set, this extended image set could obtain better performance.

3.2 Training on one object

In this experiment we seeded the recognition system with a single image of the sports-car and then iteratively found the best match to the current representation over the entire corpus and rebuilt the sports-car representation incorporating the new image. We stopped the procedure the first time a non-sports-car image was attracted. The overall procedure is essentially a minimum spanning tree algorithm.

The first 32 images attracted to the growing representation were images of the sports-car; and these represented over half of the 52 sports-car images in the entire corpus. Figure 2 shows the tree by which these $1 + 32 = 33$ views (62.2% of the sports-car views) attracted each other to the growing representation. The 52 sports-car images in the corpus represent one hemisphere, and are represented by circles on the polar coordinate system in the figure. Dark circles represent images attracted to the representation prior to the first false match. The double circle represents the seed. Arrows show the topology of the growth process. The incorporated images represented a coherent patch of the viewing sphere, and spanned over 120 degrees. The attraction process generally operated between close geometric neighbors, with one exception, where an image from the same elevation, but separated by 180 degrees in azimuth was attracted. This is due to the rough front/rear symmetry of the sedan.

Figure 3 shows the seed image and some of the sports-car images attracted to the representation (indicated by a,b,c in Figure 2). It also shows the non-sports-car image attracted at step 33 that stopped the growth process. The image is actually an odd view of a toy-rabbit block, but one that interestingly, looks a bit like the sports-car.

The other objects had somewhat similar results: the cup attracted 50.9%, the toy-bear 61.3%, the toy-rabbit 31.1%, the plane 48.1% and the fighter 61.3% of their total number of views. The toy-rabbit’s low percentage is due to its odd shape, that makes it look rectangular from the side. Since the plane and the fighter are similar, the first incorrect image attracted by the plane class was a fighter, and the first incorrect image attracted by the fighter class was a plane. In both cases the first image attracted from a different class occurred after 61.1% of the 212 plane and fighter images were attracted.

Next we performed the same experiment on the extended image set. This reduced the number of isolated views, and thus the learning performance increased in general, as can be seen in Table 1.

| object | performance on initial set | performance on extended set |
|------------|----------------------------|-----------------------------|
| cup | 50.9% | 80.1% |
| toy-bear | 61.3% | 80.6% |
| sports-car | 62.2% | 54.2% |
| toy-rabbit | 31.1% | 34.4% |
| plane | 48.11% | 54.3% |
| fighter | 61.3% | 63.3% |

Table 1: Percentage of images attracted by each object before the first incorrect classification

The biggest improvement can be seen in the case of the cup and the toy-bear where increasing the sampling from every 20° to every 15° decreased significantly the number of

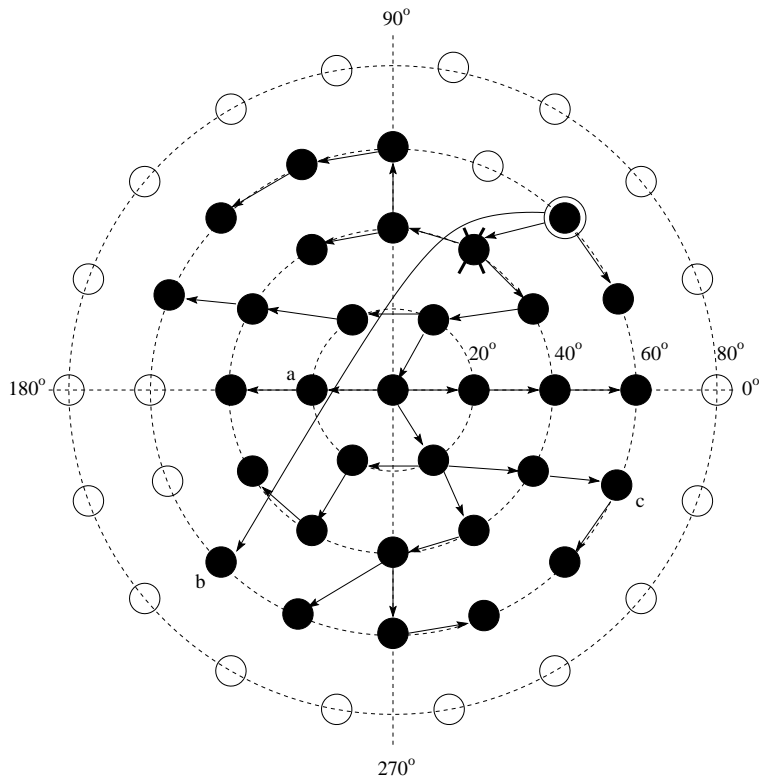


Figure 2: Tree by which training images were attracted to representation during growth process. The seed image is marked with a circle, while the image that attracted the first incorrect image is marked with an X.

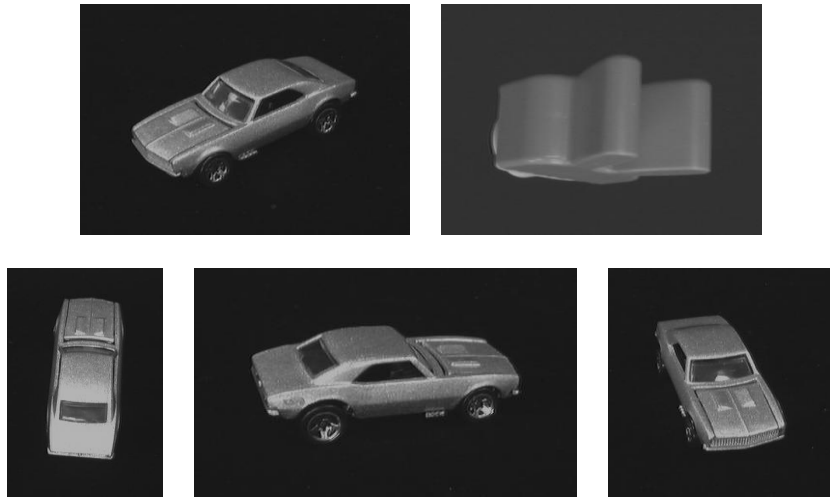


Figure 3: Top: Seed image of sports-car for propagation experiment and terminating non-car image. Bottom: car images, a,b, and c, attracted to the representation during the experiment.

isolated views. For the sports-car the sampling has to be increased even more. By adding a few additional views at some intermediate elevations we managed to increase the performance to 75.5%. These results suggest that a further increase in the sampling rate could push the performance close to 100%. This increase can be done in an adaptive fashion, so that those areas of the viewing sphere that contain more difficult views are sampled more densely than areas containing easier views.

3.3 Training on several objects

After the first experiment we trained the system on the six objects in parallel. We seeded the system with one view of each object and iteratively found the best match to the current representation over the entire corpus. The first incorrect classification occurred at image 231. The performance started degrading after that, reaching an overall performance of 60%, i.e. 60% of the images in the database were correctly classified and thus attracted to the representation of the correct model.

The evolution of the score of the best match, as well as the class that attracted each image, can be seen in Figure 4. The match score is a number of arbitrary units that is assigned by the object recognition system to every object match and gives a measure of the goodness of the match.

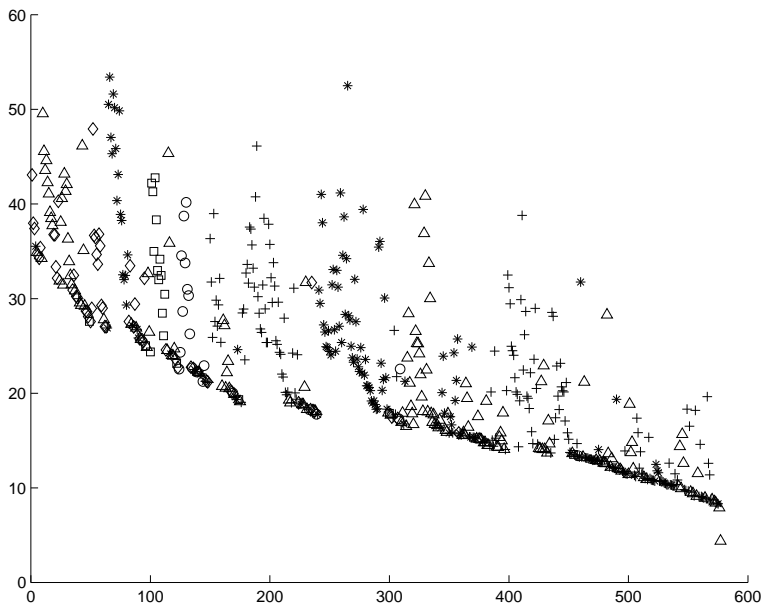


Figure 4: Score and class of the best match with one seed image per object

Similarly to the case of the sports-car, the images attracted to the object representations are neighboring and symmetrical views of the views already in the database. The recognition score is higher when the view is closer to an already known view. The images that are attracted last are the “pathological” views of objects, i.e. views that are difficult to recognize even for humans, such as the side views of the planes and the toy-rabbit, and the top and bottom views of the cup. These are also the views that get misclassified more easily.

The misclassifications were due to similar views of different objects, such as the plane and the fighter plane and the toy-rabbit and the sports-car. The initial incorrectly classified views attracted further incorrect images to the object representation. The most common misclassifications were planes classified as fighter planes and fighter planes classified as planes. This is not surprising, since these two objects are similar.

The attraction of one view to the database triggers the attraction of other views of the same object (see Figure 4). This groups the objects into several clusters. Usually there is one big cluster centered around a characteristic view of the object, containing a big part of the similar views, as well as smaller clusters centered around views that are more difficult to associate with the object’s identity. In order to get a complete and correct representation of each of the objects, the clusters describing one object should grow into a single cluster.

Each object has an associated view graph, such as the one in Figure 2. Each view can attract only a limited number of views to the representation, usually the neighboring views and sometimes some symmetric views. Thus different views attract different clusters, and eventually we want to group these clusters together.

This behavior is similar to that of spreading activation networks [5]. These networks consist of a set of nodes representing various potentially active states, with weighted links between them. The weights determine how much the activation of a given node directly affects others. In a typical application some nodes are initially activated by external inputs, and then these nodes cause others to become active. In such a network there is a phase transition that takes place when the parameters reach a certain value, where the topology of the network suddenly changes from small isolated clusters to a giant one containing very many nodes. The existence of these giant clusters allows the activation to reach arbitrarily remote regions of the network.

This phenomenon was first displayed in the seminal work of Erdős and Rényi [1] on random graphs, who showed that, for many properties Q , the limit probability that a random graph has property Q exhibits a sharp increase at some critical value of the edge probability.

Hogg and Kephart [3] applied this theory for the case of information-intensive categorization tasks. They showed that the performance of algorithms used in solving such problems changes abruptly as the number of categorization classes, the variability of the input features, and the algorithm quality are varied. Under the proper conditions, a classification algorithm which performs poorly in small problems can perform extremely well for larger problems.

Based on these results, we started varying different parameters of our algorithm. An easy way to deal with separate clusters is to use several different seed images. This way even if the images will be attracted to different clusters, they will be labeled the same way. To see how the number of seed images affects system performance, we seeded the system with two, four and eight images per object. We obtained a significant improvement when using two seed images (an overall final performance of 82%). This improvement is probably due to the fact that flattened objects have two large view clusters, and by using two seed images we were able to attract both clusters. This phenomenon suggests the need for variable sampling of the viewing sphere, with more images in the areas with difficult, “pathological” views.

A slight further improvement was obtained using four seed images per object (overall final performance was 86%). Increasing the number of seed images to eight didn’t make a difference. A comparison of how the performance of the system changes with the number of

images attracted to the representation in the case of 1, 2, 4 and 8 seed images per object can be seen in Figure 5.

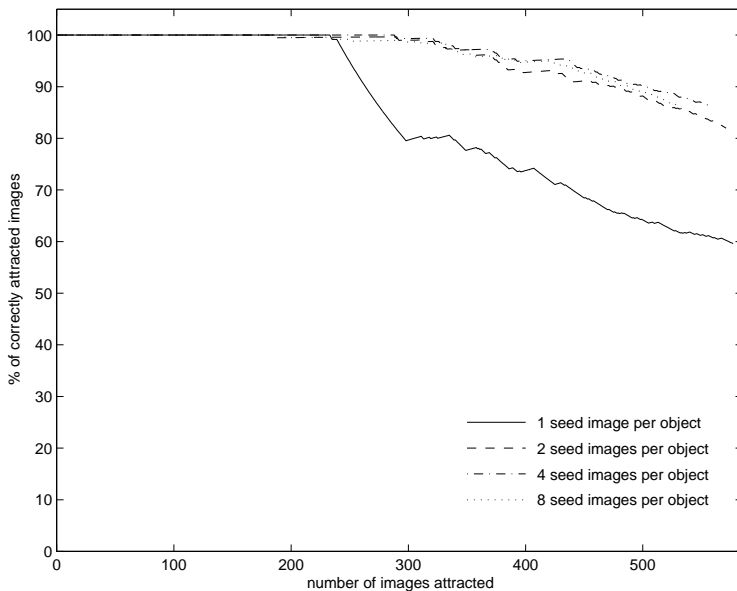


Figure 5: Performance of the learning system with 1, 2, 4 and 8 seed images per object

A better performance can be achieved by an interactive system that asks the user about the identity of images that could not be classified. This way the number of clusters can vary depending on the complexity of the object. If two separate clusters describe the same object, they will be labeled as such by the user. In the end all clusters belonging to the same object will be used in the object representation.

Images that get recognition scores above a fixed threshold are added to the object representation automatically. If none of the images obtained a high enough score during one iteration, the system asks the user about the identity of the object in the image that got the highest score. This is usually the image that has the most information and hence it might be able to attract more images to the representation.

We ran this interactive version on our corpus using 1 seed image per object and setting the threshold to 18. We had only 3 incorrect classifications (this amounts to 0.5% of the total number of images), but the number of images that had to be classified by the user was too high (202 images, 35%). A higher threshold reduces the number of incorrect classifications to 0 at the cost of increasing the number of images classified by the user, while a lower threshold increases the number of incorrect classifications.

Figure 6 shows the score and class of every image that was added to the representation.

As the number of images in the database increases, the recognition scores get lower. This is due to the fact that some of the views are more unique than others and they cannot be matched as easily to views already in the database. Since our algorithm picks the images with highest recognition scores, the views that are more difficult to match will be added later. The scores decrease also because in our object recognition system evidence coming from features that are more common in the database is lower, hence the scores will drop as

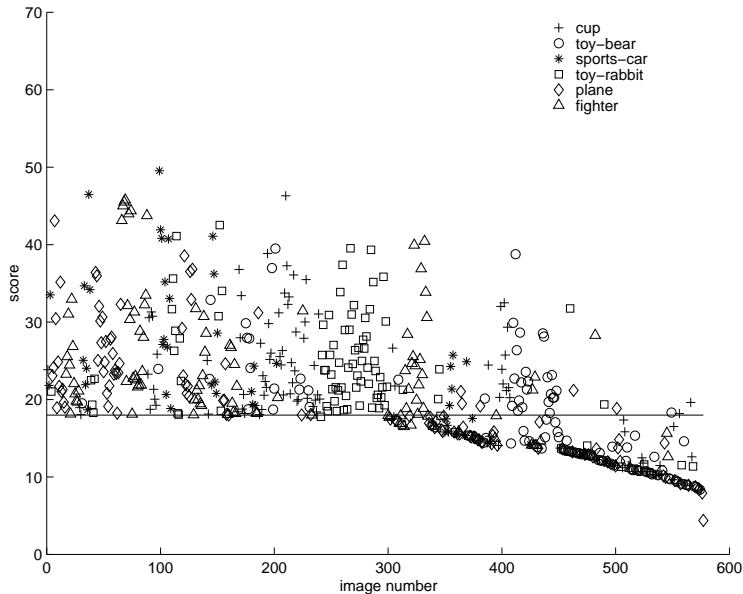


Figure 6: Score and class of the best match in the interactive version

the size of the database increases.

To solve this problem we decreased the value of the threshold used in the learning process with the number of views already attracted to the representation. The number of incorrectly classified images grew from 3 to 27 (5% of the total number of images). On the other hand the number of images that had to be classified by the user decreased to 137 (23% of the total number of images). A threshold whose value would be better linked to the way the score decreases with the increase of the database size would obtain even better results.

The number of clusters can be reduced by reducing number of isolated views. This can be done by using more intermediate images between odd views of the object. To see how this problem can be fixed, we performed some experiments on our extended image set. We ran the interactive version of our learning process and again we first used a fixed threshold (set to 20). This time there were only 4 incorrectly attracted views (0.38% of the total number of images). Out of these, 3 were views of fighters classified as planes. The number of images that had to be classified by the user was 240. This represents 23% of the total number of images, a 12% decrease from the smaller set of images. Figure 7 shows the score and class of every image that was added to the representation.

As in the case of the smaller database, the match score decreases with the number of images attracted to the representation. By decreasing the threshold with the number of images attracted, we were able to further reduce the percentage of images that had to be classified by the user to 20.6%, while the number of incorrectly classified images stayed the same.

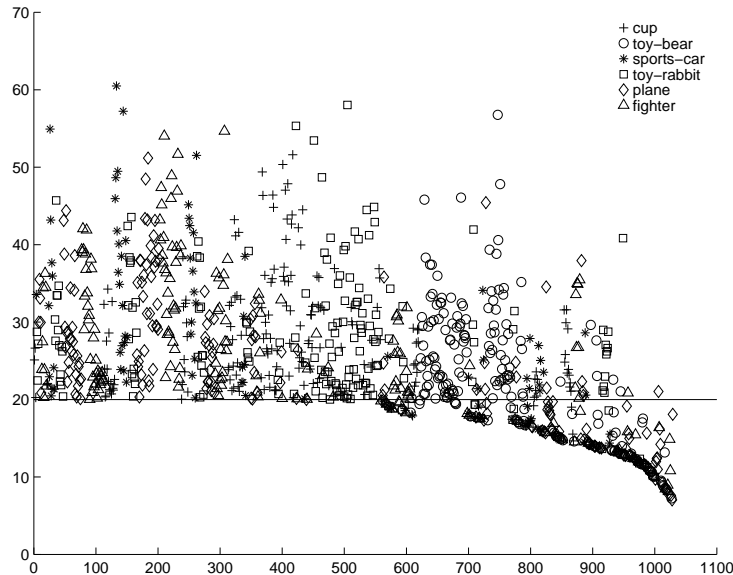


Figure 7: Score and class of the best match for the extended corpus

4 Conclusions and Future Work

Using our recognition system as-is we were able to use simple algorithms to construct representations from a clean, but unlabeled training dataset. By simply seeding the recognition system with one or more images of each object and then iteratively adding to the current representation the best match over the entire corpus, we were able to build a large part of the training database.

We showed that the views that are added to the representation come in clusters. The goal is to obtain a single cluster for each object that will contain all the views of the object. The process is similar to the phase transition phenomenon that shows up in random graphs. The cluster size can be increased by extending the set of views through an increase of the sampling rate, or the use of a variable sampling density, higher in the areas with more difficult views. Another solution is to introduce a small amount of user supervision that allows the user to give the identity of objects whose recognition score was not high enough. The threshold can be set such that false matches are not attracted at all, but in this case not all the images will be attracted.

While the results presented in this report are promising, they are preliminary and there is a lot of room for improvement. So far we used the first hypothesis given by our recognition system. Better results can be obtained by using the verification phase, especially for the images with low recognition scores. Another way of improving performance would be to guess the geometry of objects and generate hypothetical distorted views based on the views already attracted to the representation. This would eliminate the need to increase the sampling rate. Such a hypothetical view could be generated by projecting a known view onto a sphere or other generic 3D shape, rotating it, and then projecting it back to a plane.

Our next goal is to build a system that can be trained from cluttered images. The performance of our object recognition system is such that we believe this is possible by adding

some additional structure. This structure could come both through improved performance of the recognition system, and through more sophisticated training algorithms.

References

- [1] P. Erdős and A. Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Science*, 5:17–61, 1960.
- [2] W. Grimson and D. Huttenlocher. On the sensitivity of the hough transform for object recognition. *IEEE Trans. PAMI*, 12(3):255–274, 1990.
- [3] T. Hogg and J. O. Kephart. Phase transitions in high-dimensional pattern classification. *Computer Systems Science and Engineering*, 5(4):223–232, October 1990.
- [4] C. Huang, O. Camps, and T. Kanungo. Object recognition using appearance-based parts and relations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 878–884, San Juan, Puerto Rico, June 1997.
- [5] B. A. Huberman and T. Hogg. Phase transitions in artificial intelligence systems. *Artificial Intelligence*, 33:155–171, 1987.
- [6] B. Mel. Seemore: Combining color, shape, and texture histogramming in a neurally-inspired approach to visual object recognition. *Neural Computation*, 9:777–804, 1997.
- [7] H. Murase and K. Nayar. Visual learning and recognition of 3-d objects from appearance. *Int. Journal of Computer Vision*, 14(1):5–24, 1995.
- [8] R. Nelson and A. Selinger. A cubist approach to object recognition. In *Proc. International Conference on Computer Vision (ICCV98)*, pages 614–621, Bombay, India, January 1998.
- [9] R. Nelson and A. Selinger. Large-scale tests of a keyed, appearance-based 3-d object recognition system. *Vision Research*, 38(15-16):2469–88, August 1998.
- [10] T. Poggio and S. Edelman. A network that learns to recognize three-dimensional objects. *Nature*, 343(1):263–266, 1990.
- [11] R. Rao and D. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence*, 78:461–505, 1995.
- [12] B. Schiele and J. Crowley. Object recognition using multidimensional receptive field histograms. In *Proc. Fourth European Conference on Computer Vision*, pages 610–619, 1996.
- [13] C. Schmid and R. Mohr. Combining greyvalue invariants with local constraints for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 872–877, San Francisco, CA, June 1996.

- [14] A. Selinger and R. Nelson. A perceptual grouping hierarchy for appearance-based 3d object recognition. *Computer Vision and Image Understanding*, 76(1):83–92, October 1999.
- [15] Z. Wang and J. Ben-Arie. Generic object detection using model based segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 428–433, Fort Collins, CO, June 1999.