Local Feature View Clustering for 3D Object Recognition

David G. Lowe
Computer Science Department
University of British Columbia
Vancouver, B.C., V6T 1Z4, Canada
lowe@cs.ubc.ca

Abstract

There have been important recent advances in object recognition through the matching of invariant local image features. However, the existing approaches are based on matching to individual training images. This paper presents a method for combining multiple images of a 3D object into a single model representation. This provides for recognition of 3D objects from any viewpoint, the generalization of models to non-rigid changes, and improved robustness through the combination of features acquired under a range of imaging conditions. The decision of whether to cluster a training image into an existing view representation or to treat it as a new view is based on the geometric accuracy of the match to previous model views. A new probabilistic model is developed to reduce the false positive matches that would otherwise arise due to loosened geometric constraints on matching 3D and non-rigid models. A system has been developed based on these approaches that is able to robustly recognize 3D objects in cluttered natural images in sub-second times.

1. Introduction

There has recently been considerable progress in developing real-world object recognition systems based on the use of invariant local features [12, 6]. The local features are of intermediate complexity, which means that they are distinctive enough to determine likely matches in a large database of features but are sufficiently local to be insensitive to clutter and occlusion. Such features can be densely sampled over the image, clustered with a Hough transform, and verified with model fitting, leading to efficient and robust recognition in complex real-world scenes.

The existing work in this area has been based upon taking single training images of objects to be recognized and storing their features in a database for future recognition. The local feature approach can be made invariant to image rotation, translation, and scaling, but can only tolerate moderate object rotation in depth (typically about 20 degrees in each direction from the training view). One approach to generalizing to full 3D recognition might be to simply store training images acquired around the view sphere and select the best match. However, this means that new views may have features matching any of several nearby training images without any ability to integrate the information. As importantly, robustness can be greatly improved by combining features from multiple images taken under differing conditions of illumination or object variation, so that each view model contains many more of the features likely to be seen in a new image.

This paper describes an approach to combining features from multiple views to provide for full 3D object recognition and better modeling of object and imaging variations. The feature combinations are performed by measuring the closeness of the geometric fit to previous views, and views that are similar are combined into view clusters. For nearby views that are not combined, matching features are linked across the views so that a match in one view is automatically propagated as a potential match in neighboring views. The result is that additional training images continue to contribute to the robustness of the system by modeling feature variation without leading to a continuous increase in the number of view models. The goal is to eventually use this approach for on-line learning in which object models are continuously updated and refined as recognition is performed.

Another possible approach to the problem of 3D object recognition would be to solve for the explicit 3D structure of the object from matches between multiple views. This would have the advantage of leading to a more accurate fit between a rigid model and the image, leading to more accurate determination of pose and more reliable verification. However, the approach given in this paper has the advantage of not making rigidity assumptions, and therefore being able to model non-rigid object deformations. It also is able to perform recognition starting with just single training images, whereas a 3D model approach would likely require at least several images for an accurate 3D solution. It is likely that the ultimate performance would be achieved through a combination of these methods, but we show that view clustering is sufficient in many cases.

The view clustering approach allows for substantial variation in feature position during matching to account for 3D view change as well as non-rigid object variation. One consequence is that the final least-squares solution for model parameters is less effective at discarding false positive sets of feature matches than would be the case for a tightly constrained solution. Therefore, this paper develops a new probabilistic model for determining valid instances of recognition that has proved successful for these less-constrained models.

2. Related research

There is a long history of research in object recognition that has modeled 3D objects using multiple 2D views. This includes the use of aspect graphs [4], which represent topologically distinct views of image contours; eigenspace matching [8], which measures distance from a basis set of eigenvalue images; and histogram matching [11, 14] which summarize image appearance with histograms of selected properties. The work in this paper follows most closely from [10], in which the appearance of a set of images was modeled as a probability distribution, which in turn was represented as a conjunction of simpler distributions of independent features. This paper uses a different type of feature that provides more specific matches to a model database, which allows for a simpler and much more efficient model representation.

Another approach has been to use linear interpolation between edge contours that have been matched between 3 views under an orthographic viewing assumption [17]. While this can produce more accurate geometric constraints for edge contours of rigid objects, it cannot handle non-rigid objects and does not incorporate the many features that do not match between all 3 views.

3. Feature detection and matching

To allow for efficient matching between models and images, all images are first represented as a set of SIFT (Scale Invariant Feature Transform) features, which have been described in detail in earlier work [6]. Each SIFT feature represents a vector of local image measurements in a manner that is invariant to image translation, scaling, and rotation, and partially invariant to changes in illumination and local image deformations. A typical image will produce several thousand overlapping features at a wide range of scales that form a redundant representation of the original image. The local and multi-scale nature of the features makes them insensitive to noise, clutter and occlusion, while the detailed local image properties represented by the features makes them highly selective for matching to large databases of previously viewed features.

The SIFT feature locations are efficiently detected by identifying maxima and minima of a difference-of-Gaussian function in scale space. At each such location, an orientation is selected at the peak of a histogram of local image gradient orientations. A feature vector is formed by measuring the local image gradients in a region around each location in coordinates relative to the location, scale and orientation of the feature. The gradient locations are further blurred to reduce sensitivity to small local image deformations, such as result from 3D viewpoint change. In summary, the SIFT approach transforms local image features relative to coordinate frames that are expected to be stable across multiple views of an object.

The size of image region that is sampled for each feature can be varied, but the experiments described in this paper all use a vector of 128 samples for each feature to sample at 8 gradient orientations over a 4 by 4 sampling region. While the size of feature vectors is considerable larger than used by other approaches, our experiments have shown that the larger vectors are useful for giving a high degree of selectivity when matching features to a large database, resulting in improved overall accuracy and efficiency. It is possible to efficiently find matches for large vectors by using a probabilistic version of the k-d tree algorithm [2].

4. View clustering

The view clustering approach can integrate any number of training images of an object into a single model. Training images from similar viewpoints are clustered into single model views. An object model consists of a set of these model views representing appearance from a range of significantly different locations around the view sphere. Matching features are linked between adjacent model views to allow for improved matching to intermediate views.

The same matching methods are used for matching training images as for doing recognition. The training images are processed sequentially in any order, and training images from different objects can be interspersed. The training images are best taken on a uniform background, as few SIFT features are produced in the uniform regions of the image, which minimizes the incorporation of spurious features in the model. Our experiments use a black background, as that also minimizes shadows cast on the background. However, spurious features do not cause major problems other than some extra computation and memory usage, so it is also possible to use training images of objects in cluttered backgrounds, as long as a single object occupies the majority of the image.

The first training image is used to build an initial model of an object. The model consists of all SIFT features extracted from the training view, along with a record of the location, orientation, and scale of each feature within that image. This initial model can be used to match and identify views of the object over a range of rotations in depth of at least 20 degrees in any direction.

The process of matching models to subsequent images uses a Hough transform approach followed by least-squares geometric verification. Each feature is matched to the neighbor in the database with the closest Euclidean distance. The matched database feature contains a record of its model view and the location of the feature within the model view, allowing a hash table entry in the Hough transform voting for a particular model view and approximate location, scale, and image orientation. This approach has been described in detail in earlier work [6].

With view clustering, the features in each model view are linked to any similar features that were matched in adjacent views. When new images are seen from intermediate views, the features may match any of several adjacent model views which could disperse the peak in the Hough transform hash table. This is avoided by following the links from each model feature to all matching features in adjacent model views and repeating the Hough transform voting process for each of these other model features. This ensures that there will be at least one model view that accumulates votes from all potentially matching features.

In our previous work on matching single training images [6], the geometric verification was performed by using an affine transform to test the fit between model and image features. However, for the purpose of view clustering, we have found that it is important to use a similarity transform instead. While an affine transform is a good model for 3D projection of planar surfaces, it provides a poor approximation for rotation in depth of more complex 3D objects. Furthermore, the affine parameters can lead to inaccurate solutions with large deformations when only a few features are matched or when some but not all parts of a model are planar.

The similarity transform gives the mapping of a model point $[x\ y]$ to an image point $[u\ v]$ in terms of an image scaling, s, an image rotation, θ, and an image translation, $[t_x\ t_y]$:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} s\ \cos\theta & -s\ \sin\theta \\ s\ \sin\theta & s\ \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

Defining $m = s\ \cos\theta$ and $n = s\ \sin\theta$ we get,

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m & -n \\ n & m \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

We can write the equation above in a linear form collecting the unknown similarity transform parameters into a vector [10]:

$$\begin{bmatrix} x & -y & 1 & 0 \\ y & x & 0 & 1 \\ & & \cdots & \end{bmatrix} \begin{bmatrix} m \\ n \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}$$

This equation describes a single feature match, but any number of further matches can be added, with each match contributing two more rows to the first and last matrix.

We can write this linear system as

$$\mathbf{Ax} = \mathbf{b}$$

The least-squares solution for the parameters **x** can be determined by solving the corresponding normal equations,

$$\mathbf{x} = [\mathbf{A^T A}]^{-1} \mathbf{A^T b}$$

which minimizes the sum of the squares of the distances from the projected model locations to the corresponding image locations.

Using this solution for **x**, we can estimate the average error, e, remaining between each projected model feature and image feature:

$$e = \sqrt{\frac{2\ ||\mathbf{Ax} - \mathbf{b}||^2}{r - 4}}$$

where r is the number of rows in matrix **A**, from which we subtract the 4 degrees of freedom of the similarity transform. The factor 2 in the numerator accounts for the fact that the squared errors in 2 rows must be summed to measure a squared image distance.

The decision of whether to cluster a new training image with an existing model view is based on comparing e to a threshold, T. We use a value T equal to 0.05 times the maximum dimension of the training image, which results in clustering views that differ by less than roughly 20 degrees rotation in depth.

As each new training image arrives, it is matched to the previous model views and one of 3 cases can occur:

1. The training image does not match any previous object model. In this case, the image is used to form a new object model.

2. The training image matches an existing model view, and $e > T$. In this case, a new model view is formed from this training image. This is similar to forming a new object model, except that all matching features are linked between the current view and the 3 closest matching model views.

3. The training image matches an existing model view, and $e \leq T$, which means the new training image is to be combined with the existing model view. All features from the new training image are transformed into the coordinates of the model view using the similarity transform solution. The new features are added to those of the existing model view and linked to any matching features. Any features that are very similar to previous ones (have a distance that is less than a third that of the closest non-matching feature) can be discarded, as they do not add significant new information.

The result is that training images that are closely matched by a similarity transform are clustered into model views that combine their features for increased robustness. Otherwise, the training images form new views in which features are linked to their neighbors.

5. Probability model for verification

One consequence of this approach to view clustering is that the matching of images to previous model views must be able to tolerate significant geometric errors in feature positions, say as much as 20% of the maximum dimension of a model. This has the additional advantage that it can allow some degree of non-rigidity in models, such as the changing expression on a face. However, this also makes the problem of final verification of a match more difficult, as there is an increased probability that mistaken matches will happen to appear in a valid configuration.

It is not sufficient to use a metric such as just the number or type of feature matches [3, 5] to determine the presence of a model under these circumstances. Schmid [13] has proposed a more complete model based on combining feature reliabilities. However, it does not take account of the varying probabilities of false matches depending on variable projected model size and image feature density. Here we give a quite different approach that takes account of the projected size of the model, the number of features in the model image region, and the accuracy of the best-fit geometric solution.

We wish to determine $P(m|f)$, where m is the presence of the model at the given pose and f is a set of k features that have been matched between the model and image.

First, we determine the number of features that are candidates for giving rise to false matches. This is done by simply counting the number of image features, n, within the projected outline of the model view, which has proved much more accurate than measures based on average feature density. For example, this takes account of the fact that highly textured image regions or large projected model sizes will have more potential false matches.

Let p be the probability of accidentally matching a single image feature to the current model pose. Then

$$p = dlrs$$

where d is the probability of accidentally selecting a database match to the current model, and l, r, and s are the probabilities of satisfying the location, orientation, and scale constraints respectively. As we are matching each feature to its closest neighbor in the database of models, d is given by the fraction of the database occupied by features from this model view. For the location constraint, we are only considering the n features that lie within the bounds of the projected model view, so we need consider only the additional constraint on location imposed by the pose solution. For our examples, we constrain location to within a range of 20% of the average model size in each direction, giving $l = 0.2^2 = 0.04$. Orientation is constrained within a range of 30 degrees, giving $r = 30/360 = 0.085$. The scale constraint is less effective, as keypoint scale can only be constrained within a one-octave range due to inaccuracy in scale measurements and since most keypoints are detected at the finest scales, resulting in an estimate of $s = 0.5$.

Let $P(f|\neg m)$ be the probability that the matched features, f, would arise by accident if the model, m, is not present. We assume the k feature matches arose from n possible features, each of which matches by accident with probability p. Therefore, we can use the cumulative binomial distribution for the probability of an event with probability p occurring at least k times out of n trials:

$$P(f|\neg m) = \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j}$$

This function can be expensive to compute from this formula for larger values of n, but an efficient method is given in [9].

To compute $P(m|f)$, we use Bayes' theorem:

$$\begin{aligned} P(m|f) &= \frac{P(f|m)P(m)}{P(f)} \\ &= \frac{P(f|m)P(m)}{P(f|m)P(m) + P(f|\neg m)P(\neg m)} \end{aligned}$$

With only an insignificant effect on the result, we can approximate $P(f|m)$ as 1, as we normally expect to see at least k features present when the model is present (for the small values of k for which this evaluation is relevant). We can also approximate $P(\neg m)$ with the value 1 as there is a very low prior probability of a model appearing at a particular pose. Therefore,

$$P(m|f) \approx \frac{P(m)}{P(m) + P(f|\neg m)}$$

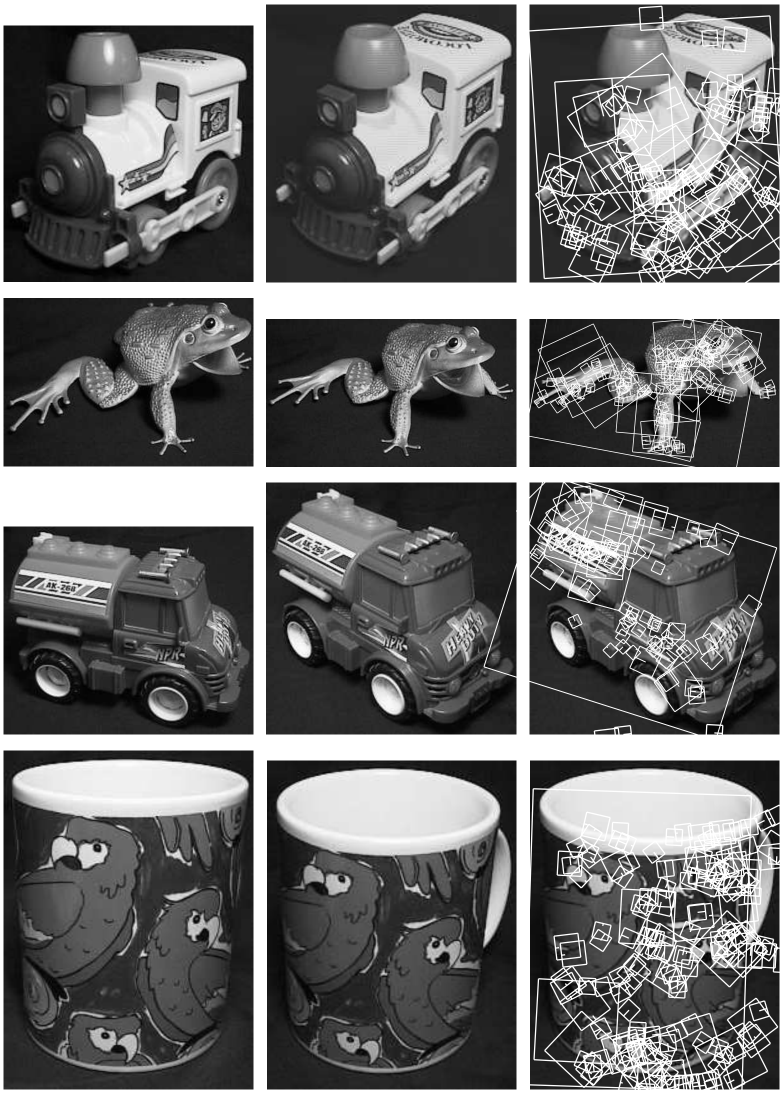

Figure 1. Pairs of model images from different views are shown on the left. The right image in each row shows the features matched between the two views on the left as superimposed on the second image. The large rectangle around the features shows the boundaries of the first image following application of the similarity transform to coordinates of the second image.

In other words, we have high confidence in the interpretation if the probability of the features matching accidentally is much lower than the prior probability of the model appearing at this pose.

It is difficult to assign a value to $P(m)$ in isolation, as the value depends on the range of poses covered by the hypothesis m relative to all possible poses. We can simplify this problem by assuming that one matching feature is used to determine the initial pose hypothesis (incorporated into m), and the remaining features are used for verification (reducing k by 1). Then, $P(m)$ is simply the probability that a single feature match is correct, which is the ratio of correctly matched features to all matched features in a typical image (about 0.01 for our cluttered images).

The final decision of whether to accept a model interpretation is made according to the relative utility of avoiding false positives or false negatives for any particular application. We use $P(m|f) > 0.95$.

6. Experimental results

The view clustering and verification approaches have proven to work well in practice. Figure 1 shows an example of some training images collected from common objects. These are taken with a handheld camera without special lighting on a black background. In this example, about 20 to 30 images were taken of each object around at least a hemisphere of viewing directions. The selection of viewpoints was very approximate, with some being quite close but an attempt being made to have none differ by more than about 45 degrees from their neighbors. Figure 1 shows a sample of two nearby views of each object. The right column of this figure shows the matched set of SIFT features from the first image overlayed on the second image. Each square shows one matched feature, with the location, size, and orientation of the square indicating the corresponding parameters for the SIFT feature. In addition, the similarity transform solution was applied to the boundaries of the first image to produce a rectangular outline shown superimposed on the second image. As can be seen, a large number of matches are found in the training phase due to the good resolution and lack of occlusion.

Figure 2 shows an image containing the modeled objects in a cluttered background. The squares correspond to the SIFT features that were matched between the model and image for each final verified match. The rectangle drawn around each match is computed by applying the similarity transform to the image boundaries of the first training image that was used to construct the matching model view. The features and boundary for each matched object are displayed with a different line thickness to visually separate the multiple object matches. The recognition is quite robust to occlusion, as can be seen by the many matching features and considering that as few

Figure 2. Recognition results using view interpolation for 3D objects showing model boundaries and image features used for matching. Each model uses a different line thickness to help separate results.

as 4 matching features are sufficient for verified recognition.

Our implementation of these methods is able to process and match each training or recognition image in about 0.8 seconds on a Pentium III 600MHz computer. About half of the time is devoted to image acquisition and feature detection, while the remaining time is used for all aspects of matching and model formation. Our implementation can run on a laptop using an inexpensive USB camera, so this approach has been tested and demonstrated on large numbers of real-world images. Recognition has proved quite robust for most objects, with the major difficulties arising for objects seen under illumination conditions that greatly differ from those in any training views, simple objects lacking in visual texture, or objects that occupy a very small part of the image. The SIFT features are insensitive to illumination change for planar surface markings, but often fail to match when features are formed by surface relief that casts different shadows under different illumination. Therefore, the ability to combine images taken under differing illumination into a single view cluster has proved valuable for robustness. We expect further improvements in the future as we develop new feature types that incorporate color, texture, and other properties.

7. Conclusions and future work

There are two major benefits of combining multiple views into a single model representation: (1) the model can be generalized across multiple 3D viewpoints and non-rigid deformations by linking features between differing views, and (2) increased robustness is obtained by combining features obtained under multiple imaging conditions into a single model view. The decision of whether to cluster images can be made by considering the residual of the geometric error following feature matching. When images are clustered, all features are placed within a common view-based coordinate system. When images are not clustered, features are explicitly linked between adjacent views to improve the reliability of matching for intermediate images.

A novel probabilistic model has been developed to provide final verification of whether a set of feature matches corresponds to an object instance. This problem was found to be particularly important due to the loosened geometric constraints on matching between approximate model views. The solution takes account of model database size, projected model size, image complexity, and accuracy of pose constraints.

Although the current system uses only a single type of local invariant feature, there are many future directions for improving robustness by incorporating a range of other feature types. Mikolajczyk & Schmid [7] have recently described an interesting new scale invariant feature selector, that may provide improved robustness at the cost of higher computation. Baumberg [1] has developed a local feature descriptor that incorporates local affine invariance with rotation invariant Fourier descriptors. Features that incorporate texture measures and color ratios in addition to image gradients can be expected to provide better matching reliability under many circumstances. Another valuable feature type would be one that incorporates figure/ground discrimination by using image properties from only one side of an edge. This would allow features to be stable near the boundaries of objects without being influenced by changes in the background. All of the different feature types could be used simultaneously, as is done in human and animal vision [16], so that each is available in those circumstances in which they provide the most reliable matches. The invariant local feature approach is almost entirely insensitive to failed matches, so it should show continuing improvements as new feature types are developed.

The current object models consist of multiple views, each of which is formed from a cluster of training images. This works well for object detection, but does not solve for accurate 6 degree-of-freedom pose and other object parameters. Further improvement would result from using multiple matched training images to explicitly solve for the 3D structure of a rigid model [15]. This would provide for accurate pose solutions and would improve robustness for small numbers of feature matches by enforcing more accurate geometric constraints. However, it may not be as suitable for modeling non-rigid changes or performing recognition from only one or two views, so such a solution would need to be combined with the current approach rather than replace it.

Acknowledgments

I would like to thank Matthew Brown, Krystian Mikolajczyk, Stephen Se, and Jim Little for their many useful comments and suggestions regarding this work. This research was supported by the Natural Sciences and Engineering Research Council of Canada and by the Institute for Robotics and Intelligent Systems of the Networks of Centres of Excellence.

References

[1] Baumberg, Adam, "Reliable feature matching across widely separated views," *Conference on Computer Vision and Pattern Recognition*, Hilton Head, South Carolina (June 2000), pp. 774–781.

[2] Beis, Jeff, and David G. Lowe, "Shape indexing using approximate nearest-neighbour search in high-dimensional spaces," *Conference on Computer Vision and Pattern Recognition*, Puerto Rico (1997), pp. 1000–1006.

[3] Grimson, Eric, *Object Recognition by Computer: The Role of Geometric Constraints*, The MIT Press, Cambridge, Mass. (1990).

[4] Koenderink, J.J., and A.J. van Doorn, "The internal representation of solid shape with respect to vision," *Biological Cybernetics*, **32** (1979), pp. 211–216.

[5] Lowe, David G., "Three-dimensional object recognition from single two-dimensional images," *Artificial Intelligence*, **31,** 3 (1987), pp. 355–395.

[6] Lowe, David G., "Object recognition from local scale-invariant features," *International Conference on Computer Vision*, Corfu, Greece (September 1999), pp. 1150–1157.

[7] Mikolajczyk, Krystian, and Cordelia Schmid, "Indexing based on scale invariant interest points," *International Conference on Computer Vision*, Vancouver, Canada (July 2001), pp. 525–531.

[8] Murase, Hiroshi, and Shree K. Nayar, "Visual learning and recognition of 3-D objects from appearance," *International Journal of Computer Vision*, **14,** 1 (1995), pp. 5–24.

[9] Press, W. H., *et al.*, *Numerical Recipes in C*, Cambridge University Press (1988), p. 182.

[10] Pope, Arthur R., and David G. Lowe, "Probabilistic models of appearance for 3-D object recognition," *International Journal of Computer Vision*, 40, 2 (2000), pp. 149–167.

[11] Schiele, Bernt, and James L. Crowley, "Recognition without correspondence using multidimensional receptive field histograms," *International Journal of Computer Vision*, **36,** 1 (2000), pp. 31–50.

[12] Schmid, C., and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE PAMI*, **19,** 5 (1997), pp. 530–534.

[13] Schmid, C., "A structured probabilistic model for recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, CO (1999), pp. 485–490.

[14] Swain, M., and D. Ballard, "Color indexing," *International Journal of Computer Vision*, **7,** 1 (1991), pp. 11–32.

[15] Szeliski, R., and S.B. Kang, "Recovering 3D shape and motion from image streams using nonlinear least squares," *Journal of Visual Communication and Image Representation*, **5,** 1 (1994), pp. 10–28.

[16] Tanaka, Keiji, "Mechanisms of visual object recognition: monkey and human studies," *Current Opinion in Neurobiology*, **7** (1997), pp. 523–529.

[17] Ullman, Shimon, and Ronen Basri, "Recognition by linear combination of models," *IEEE PAMI*, **13,** 10 (1991), pp. 992–1006.