Toward Learning Visual Discrimination Strategies

Justus H. Piater and Roderic A. Grupen
Computer Science Department
University of Massachusetts
Amherst, MA 01003
{piater|grupen}@cs.umass.edu

Abstract

Humans learn strategies for visual discrimination through interaction with their environment. Discrimination skills are refined as demanded by the task at hand, and are not a priori determined by any particular feature set. Tasks are typically incompletely specified and evolve continually. This work presents a general framework for learning visual discrimination that addresses some of these characteristics. It is based on an infinite combinatorial feature space consisting of primitive features such as oriented edgels and texture signatures, and compositions thereof. Features are progressively sampled from this space in a simple-to-complex manner. A simple recognition procedure queries learned features one by one and rules out candidate object classes that do not sufficiently exhibit the queried feature. Training images are presented sequentially to the learning system, which incrementally discovers features for recognition. Experimental results on two databases of geometric objects illustrate the applicability of the framework.

1. Introduction

The extraction of useful information from visual data is a hard problem. This is true not only for artificial systems: About half of the human brain is devoted directly or indirectly to vision [14]. While the precise mechanisms underlying human visual perception are still poorly understood, there is substantial evidence that human visual learning is facilitated by a coupling of perception and action (see [6] for a thorough discussion). As we interact with our environment, we learn to pay attention to perceptual cues that are behaviorally important. For instance, we learn to recognize and distinguish individual objects and form categories on the basis of their relevance. Human ability to perceive distinctions is not primarily determined by the recognition method employed by our visual system. The converse is true: To a large extent, we can learn to make the distinctions demanded by our environment.

This work was supported in part by the National Science Foundation under grants CISE/CDA-9703217, IRI-9704530 and IRI-9503687, and by the Air Force Research Labs, IFTD (via DARPA) under grant F30602-97-2-0032.

How do humans learn recognition skills? Two principal hypotheses can be identified [10]: According to the Schema Hypothesis, sensory input is matched to internal *representations of objects* that are built and refined through experience. On the other hand, the Differentiation Hypothesis postulates that *contrastive relations* are learned that serve to distinguish among the items. Psychological evidence argues for a strong role of Differentiation learning [10, 13, 16]. What exactly the discriminative features are and how they are discovered is unclear. It appears that feature discovery is a hard problem even for humans and takes a long time to learn [6]: Neonates can distinguish certain patterns, apparently based on statistical features like spatial intensity variance or contour density. Infants begin to note simple coarse-level geometric relationships, but perform poorly in the presence of distracting cues. They do not consistently pay attention to contours and shapes. At the age of about two years, children begin to discover fine-grained details and higher-order geometric relationships. Such skills continue to grow over much of childhood.

Most work in machine recognition concentrates on Schema methods without such a developmental component. Hence, the performance characteristics of most existing machine vision systems are largely determined a priori by the design of the features and matching algorithms. Nevertheless, remarkable efforts have recently led to sophisticated statistical, texture- and shape-based features and recognition algorithms which perform impressively well on closed tasks where all training data are available at the outset [8, 12, 7, 9].

This work is not aimed at improving the recognition performance achieved by these or any other systems. In contrast, we present a method for learning *discriminative* capabilities based on an infinite feature space that can in principle express any identifiable distinction between objects. Features are learned in a simple-to-complex manner, resembling the human developmental course as outlined above. Our method is designed for open tasks where visual scenes are presented sequentially, and the number and nature of the categories to be learned are not initially known to the system.

The following section introduces the feature space. Section 3 then describes the proposed approach at a high level, and the following two sections present details of the recognition and feature discovery algorithms. Experimental results are discussed in Section 6.

2. Features

In order to learn distinctions at various levels of detail which are initially unknown, a very large feature space is required, along with a method of generating features from this space. Since it is practically impossible to make optimal use of a very large feature set [4], we employ the following two simplifying strategies [1, 2]: (1) Impose a *partial order* on the feature space that categorizes the features into various levels of structural complexity. The underlying assumption is that structurally simple features are easier to discover and have less discriminative potential than complicated features, but are still useful for some aspects of the learning problem. (2) Because exhaustive search in feature space is prohibitive, *sample* features from the feature space, beginning at the lowest level of complexity, and consider more sophisticated features as required. It can be argued that these strategies parallel those employed by human infants.

An obvious way to generate an infinite and partially ordered feature space is through combinatorics: *Primitive* features can be composed in various ways to yield *higher-order* features, which in turn can be composed. Any type of local image property can potentially serve as a primitive feature. In the context of an interactive vision system, this general framework may encompass three-dimensional or temporal cues in addition to conventional image properties.

Our system currently employs two types of primitive features: (1) An *edgel* is given by the orientation of a step edge at a given point in the image. This orientation θ is computed efficiently using the steerability property [5] of oriented Gaussian-derivative filters:

$$\tan\theta = \frac{G_1^{\pi/2}}{G_1^0}, \qquad (1)$$

where $G_1^0 = \frac{\partial}{\partial x}G(x,y)$ and $G_1^{\pi/2} = \frac{\partial}{\partial y}G(x,y)$, and $G(x,y)$ is a 2-D Gaussian. Intuitively, geometric combinations of edgels characterize aspects of shape [3]. (2) A *texel* is a vector of responses of multiple oriented Gaussian-derivative filters at various scales. At each scale and for each derivative d, a steerable basis consisting of $d+1$ filter responses at specific orientations is computed [11]. Intuitively, a texel expresses local texture characteristics. Notably, both primitives can be steered to specific orientations. This property is used to achieve invariance with respect to image-plane rotation.

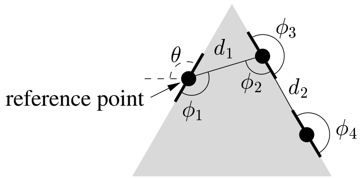

Figure 1. A geometric feature of order 3, composed of three primitives. The feature is defined by the angles ϕ and the distances d, and the orientation of this specific instance is denoted by θ. Each primitive is either an edgel or a texel.

Two intuitively expressive types of feature compositions have so far been implemented: (1) *Geometric* relations are given by the relative angles and distances between the participating lower-order features (Figure 1). As long as these are rotation-invariant, so is their geometric composition. (2) *Co-presence* asserts the presence of the participating lower-order features without making any statement about their geometric or topological relationship.

Features of any type can be composed into a co-presence feature, while only primitive and geometric features can be composed into geometric features. Note that these two types of composition constitute two extremes along a continuum. One could conceivably define a composition that asserts relaxed geometric or topological relationships between its constituents.

Features are computed at various scales, generated by successively subsampling images by factor two. This achieves a certain degree of scale invariance. Moreover, many compositions of edgels are inherently tolerant to changes in scale. For example, the arrangement shown in Figure 1 applies equally to triangles of various sizes. Another desirable property of these features is that they do not rely on explicit contour extraction or segmentation. This avoids two difficult, open problems, which should provide robustness to various kinds of image degradation.

3. Paradigm

Initially, our imaginary agent does not know anything about the visual distinctions it will need to learn, nor does it possess any recognition skills. It does, however, have the ability to search the feature space described above for useful features. As it interacts with the environment, it encounters various visual scenes and notes their different behavioral relevance. It then tries to learn features of these scenes that predict their relevance.

We believe that this general scenario captures certain aspects of childhood development. In contrast to most current approaches to machine vision, learning is inherently sequential and open-ended.

In the remainder of the paper, we disregard the interactive aspect for simplicity and discuss our system in the context of a sequential object recognition scenario. Initially, the system does not know anything about specific features or the task. Then, training images are presented one by one, and the system is asked to predict the correct class label. Whenever this fails, the system tries to discover new features that, in conjunction with previously learned features, improve its capabilities. For the purpose of sampling and evaluating new features, the system is capable of storing previously seen training images, which are then called *example images*.

The specific recognition algorithm employed in this work is not important; any of a variety of methods can be applied. Since features are sought whenever recognition fails, features are learned that work well with the specific recognition mechanism. Our preliminary implementation adopts a simple recognition algorithm, which serves to illustrate the interplay between feature selection and recognition.

4. Recognition

The idea underlying our recognition procedure is that individual features provide varying degrees of evidence in favor of some classes, and against some others. As a design choice, only the presence of a feature in an unknown image is considered evidence, not its absence. This should serve to increase robustness with respect to partial occlusion. On the other hand, in the absence of other means this implies that the system cannot uniquely discern certain classes.

We assume here that the learning procedure has provided for us a set of features for recognition, and a set of example images. The recognition procedure queries individual features in sequence, and maintains a list of candidate classes. A query of a feature either serves to rule out one or more candidate classes, or leaves the candidate list unaltered. The goal of recognition is to reduce the list of candidates to length one.

To find the best feature to query, we employ the generalized Kolmogorov-Smirnoff distance (KSD) [15], which is competitive with the best known decision tree metrics. Given a variable, it returns a cutpoint that maximizes the difference between the class-conditional cumulative distributions of this variable. The variable to be queried is the one that maximizes this metric among all variables.

Here, the variable associated with a feature f is the maximum *strength* $s_f^{\max}$ of this feature in an image. Computation of the maximum strength involves asserting the presence of the feature and measuring its strength, at each location in the image, at each scale (subsampling level). To assert a geometric feature f of order o at location (i_1, j_1), the local orientation $\theta(i_1, j_1)$ is first computed using Equation 1. We now have the location and orientation of the reference point of the feature (cf. Figure 1). The coordinates (i_k, j_k) and orientations θ_k of the other points of this feature are computed for $k = 2, \ldots, o$ using the ϕ and d values of this feature. The strength s_f is then given by the product of the strengths of the lower-order component features. The strength of an edgel is given by the response of a Gaussian-derivative filter at the desired orientation θ, which is computed using the steering equation

$$G_1^\theta = G_1^0 \cos\theta + G_1^{\pi/2} \sin\theta \qquad (2)$$

normalized by the maximum G_1 response occurring in the entire image. The strength of a texel is the maximum of zero and the normalized cross-correlation between its response vector and the pattern vector. To achieve rotational invariance, before computing the correlation the vector in question is steered to the same orientation as the pattern vector, where the orientation is computed using the first-derivative responses at the largest scale [11]. Some tolerance in the relative angles ϕ is granted through the smooth and wide peaks of the G_1^θ (a sum of two sinusoids), and some tolerance in the distance d is provided by the size of the Gaussian kernels which do not require alignment at pixel accuracy. The strength of a co-presence feature is similarly given by the product of the maximum strengths of its constituents.

Rather than asserting a feature at each point in an image, it is sufficient to consider only a small subset of salient points in the image. As long as not too many points in high-contrast areas of the image are missed, the choice of the precise saliency function is not critical. We currently employ a Canny edge detector with very low thresholds. Note also that the $s_f^{\max}$ need only be computed once for each image and can then be stored for future use.

Given a subset C of candidate classes, the best feature to query is one that maximizes the KSD among the example images of these classes. Once the best feature f^* and the corresponding cutpoint c^* are identified, the strength $s_{f^*}^{\max}$ is computed for the queried image. If $s_{f^*}^{\max} \leq c^*$, then no decision is made on the basis of this cutpoint, in accordance with the possibility that the feature f^* might actually be present but occluded or otherwise weakened in this image. In this case, the KSD for this feature is recomputed over all candidate cutpoints $c_i < s_{f^*}^{\max}$. If $s_{f^*}^{\max} > c^*$, then we identify all classes for which the majority of all example images have $s_{f^*}^{\max} \leq c^*$, and remove them from the list of candidate classes.

Notice that while up to here our procedure was analogous to conventional decision trees, this last step constitutes a major simplification. In decision trees, candidate classes are split across the current cutpoint, whereas we maintain them in their entirety. This assumes that the class-conditional densities are unimodal and reasonably well separated, which is not generally true in practice. On the other hand, this procedure effectively avoids overfitting which in the case of a potentially infinite, dynamic feature set is more critical than

Table 1. Recognizing a novel image I.

1.	$C :=$ {all classes}
2.	$f^* :=$ best feature, c^* the corresponding cutpoint. If no features left, return C.
3.	If $s_{f^*}^{\max}(I) \leq c^*$, re-evaluate this f^*, ensuring $c^*_{\text{new}} < s_{f^*}^{\max}(I)$. Go to step 2.
4.	$C := C \setminus$ { all classes for which the majority of all example images has $s_{f^*}^{\max}(I_i) \leq c^*$}
5.	If $\lvert C \rvert > 1$, go to step 2. Otherwise, return C.

Table 2. Learning a novel image I.

1.	If I is recognized correctly, stop.
2.	Add I to the example images and compute the $s_f^{\max}(I)$.
3.	If I is recognized correctly, stop. Else, note C and $\mathsf{ksd}(f_{\text{old}})$ at the failing recognition step.
4.	Generate a candidate f_{new}.
5.	If $\mathsf{ksd}(f_{\text{new}}) > \mathsf{ksd}(f_{\text{old}})$ and $s_{f_{\text{new}}}^{\max} > c_{\text{new}}$, add f_{new} to the set and go to step 3.
6.	If the maximum number of new features is reached, stop; else go to step 4.

in typical classification problems involving a fixed set of features: In our problem, almost any two training images can be distinguished by some feature, which would have a devastating effect on the generalization properties of the resulting classifier. Rather, we want to force our system to learn better features for which the assumption of unimodality is as close to true as possible. This restricts the applicability of the current system to objects that possess characteristic features visible in any pose, a property shared by most appearance-based recognition schemes which is commonly overcome by training on multiple object poses.

This procedure is iterated until one of the following situations occurs: (1) There is only one candidate class left, which is then returned as the classification. (2) There is no candidate left, which means the system is totally unable to make any statement about the classification. (3) The feature set is exhausted. In this case, the remaining list of candidate classes is returned as possible classifications. In cases 2 and 3, and in case 1 if the classification is wrong, the recognition has failed. In that case a new feature is sought that solves the problem, as described in the next section. The entire recognition procedure is summarized in Table 1. Note that all KSDs and decision thresholds can in principle be precomputed, which allows the construction of a special type of decision tree. In this case, the expected time taken to recognize an unknown image is logarithmic in the number of classes, and does not directly depend on the number of features or the number of stored example images.

5. Feature discovery

If recognition of an object fails, the image is added to the set of example images, and the $s_f^{\max}(I)$ are computed for all f. It is then run through the recognition procedure again because some KSDs may have changed to our advantage. Only if recognition fails again, a new feature is sought.

What are the properties required of the new feature? We note that a recognition can fail for one of two reasons: Either the correct class is ruled out at some stage during the recognition process, or the system runs out of suitable features and returns a set of possible class labels which contains the correct one.

In the first case, we want to find a feature f_{new} to be employed by the recognition procedure in place of the feature f_{old} that previously ruled out the true class. Thus, the KSD achieved by f_{new}, $\mathsf{ksd}(f_{\text{new}})$, needs to be greater than $\mathsf{ksd}(f_{\text{old}})$ among the subset C of all classes still under consideration at this stage in the recognition process. In the second case, C is the set of classes returned by the recognition procedure, and $\mathsf{ksd}(f_{\text{old}})$ is taken to be 0.5, which in the case of two classes corresponds to a split that places the majority of one class below the cutpoint, and the majority of the other class above. In both cases, the feature must be present in image I to a degree stronger than the cutpoint associated with f_{new}, i.e. the cutpoint c_{new} is chosen to satisfy $s_{f_{\text{new}}}^{\max}(I) > c_{\text{new}}$.

Such an f_{new} is sought by randomly sampling features from image I. This sampling proceeds in stages: First, some number of new order-2 geometric edgel features and order-1 texels are generated by randomly choosing points from among the salient pixels in I, and noting the two angles ϕ_i and the distance d, if applicable. To keep the features local, the distance between two sampled edgels is limited. Next, all existing geometric features (i.e. those previously learned and those just sampled) are augmented to higher-order geometric features. This is done by sampling a new primitive – edgel or texel – and noting the resulting ϕ_i and d with respect to the reference point of the parent feature. At the third stage, randomly chosen pairs of previously learned or newly sampled candidate features are composed into co-presence features. The sampling process is terminated once a feature f_{new} achieves $\mathsf{ksd}(f_{\text{new}}) > \mathsf{ksd}(f_{\text{old}})$ with a cutpoint $c_{\text{new}} < s_{f_{\text{new}}}^{\max}(I)$, or after a maximum number of composition steps is completed without success.

If a suitable feature is found, it is added to the set, and the current training image is again run through the recognition procedure. The properties of the new feature guarantee that either of the following occurs: (1) The new feature is chosen at the stage that previously failed during the recognition process, and the correct class is not ruled out at this stage. (2) It is chosen at some earlier stage during the recognition process. If the recognition fails again, the feature sampling process iterates. The feature learning procedure is summarized in Table 2.

For a brief look at the time complexity, first note that the feature composition process involves iterating over all pre-existing features and newly sampled candidate features, say n_f in total. Computation of $\mathsf{ksd}(f_{\text{new}})$ requires processing each example image in each class under consideration, which on average is proportional to the total number n_I of accumulated example images. Therefore, learning one new feature has a time complexity on the order of $2^a n_f n_I$, where a is a small constant giving the maximum number of composition steps. Since the number of pre-existing features is directly related to the number n_I of accumulated example images, finding one new feature takes time proportional to n_I^2. Clearly this is not acceptable for large-scale recognition problems. At the very least, suitable heuristics for reducing both factors n_f and n_I in the complexity term need to be identified.

6. Examples

To illustrate the operation of our system, we trained it on two simple supervised object recognition tasks, each containing example views of simple geometric objects. In one task, the database consisted of eight synthetic objects[1], each of which was rendered in high quality at 15 different views, covering 40 horizontal and 20 vertical degrees of the viewing sphere (Figure 2). For the other task, low-quality images were taken of real geometric objects (Figure 3). There were 18 views of a sphere, 19 views of a cone in various positions, and 16 random views of a cube. The images of the class "sphere" included spheres of two different sizes, and the images of the class "cube" contained two cubes that differed in size.

The learning system was trained on each task as described above. The images of the training set were iteratively presented to the system in random order, until either the system had learned the training set perfectly, or until no feature was found during an entire pass through the training set even though there were some misclassifications. To learn a new feature, first up to 10 new features were sampled (individual texels or pairs of edgels). Then, the set of all geometric pre-existing and new candidate features was augmented by one edgel or texel. Finally, up to the same number of co-presence features was generated. Table 3 shows the results obtained by 10-fold stratified cross-validation. In all test cases, the recognition procedure returned a single class label.

The synthetic objects were learned almost perfectly. The real-object task was much harder because objects varied in size and were presented in entirely random positions, which forced the system to find largely pose-invariant features. Our algorithm tends to mistake spheres for cones because spheres lack features that distinguish them from a cone lying down, revealing its circular base. Three out of the four misclassified cone images are accidental views that hide the characteristic conic shape.

[1]http://www.cis.plym.ac.uk/cis/levi/UoP_CIS_3D_Archive/8obj_set.tar

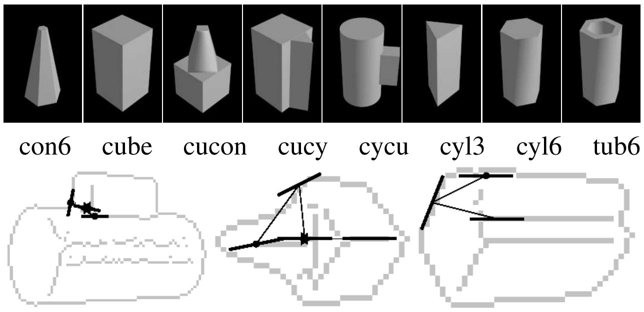

Figure 2. The synthetic-object task: Example views and examples of features learned.

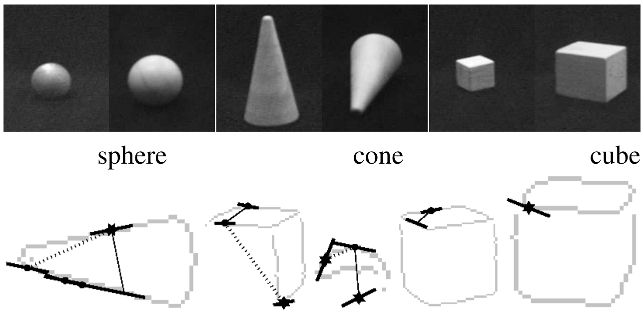

Figure 3. The real-object task: Example views and examples of features learned.

Figures 2 and 3 include some examples of features found during learning. The gray lines indicate the salient points used for sampling new features. Texels are marked by a small star, geometric relations by a solid line, and co-presence connections by a broken line.

It took between two and five passes through the training set to learn the training images. Sometimes the training set was not perfectly learned, which happened in two (out of ten) runs on the synthetic task, and in five runs on the real task. The number of accumulated example images varied between 31 and 49 on the synthetic and between 13 and 32 on the real task. For both tasks, between 8 and 23 features were learned per run. However, only about half of them were actually consulted on some training or test image. In other words, the other half had been superseded by a better feature at later stages of training.

7. Conclusions and future work

Adaptive, interactive agents – whether biological or artificial – benefit from learning those visual distinctions that turn out to be relevant for their tasks or behaviors. This learning process is inherently sequential, never complete, and unknown at the outset. We have presented a framework for progressive learning of such open-ended visual discrimination tasks. It is based on a combinatorial feature space of

Table 3. Confusion matrices summarizing the cross-validated test-set performances. The overall proportion of correct recognitions was 0.96 on the synthetic objects, and 0.83 on the real objects.

	classification results on synthetic objects:								sums:
	con6	cube	cucon	cucy	cycu	cyl3	cyl6	tub6	
con6	15								15
cube		15							15
cucon			15						15
cucy		1		14					15
cycu			1		14				15
cyl3						14		1	15
cyl6			1				13	1	15
tub6								15	15
sums:	15	16	17	14	14	14	13	17	120

	results on real objects:			sums:
	sphere	cube	cone	
sphere	14	1	3	18
cube		15	1	16
cone		4	15	19
sums:	14	20	19	53

potentially infinite size. Our framework is general enough to incorporate any type of localized spatial or temporal image property as feature primitives, and a variety of means for composing them into higher-order features.

A structural simple-to-complex partial ordering of the feature space permits feature search in polynomial time. While simple-to-complex feature sampling is not generally optimal with respect to any meaningful objective, this heuristic is intuitively pleasing in that it prefers simplicity over complexity. Assuming that most distinctions between object classes can be expressed in terms of low-order features, simple-to-complex sampling expends most effort in those areas of the feature space where success is most likely to occur. Psychological data indicate that such a strategy may also be followed during the human developmental course.

On the downside, our current feature sampling method is limited in that the search in feature space is essentially blind. It is only guided by the requirement that a new feature be present in the scene to be learned, and by simple locality heuristics. The identification of more focused search methods would lead to significant improvements in performance. Ideally, a system would learn heuristics or optimized systematic strategies for discovering useful features.

The most serious limitation of our current system is the simplicity of the recognition procedure. It limits the general applicability of our approach in two ways: (a) Sequential query of features and hard decisions against candidate classes at each step makes strong assumptions about the nature of the task, which are unrealistic in most interesting problems; (b) The decision tree approach requires that every newly sampled candidate feature be evaluated on each stored example image. Therefore, our recognition procedure does not scale well to large problems.

An ideal recognition procedure permits sequential accumulation of evidence while avoiding hard decisions. Information theoretic measures can guide a purposive search for evidence. The evaluation of the utility of newly sampled features can in most cases be approximated. Our current work addresses these goals in a system that employs a Bayesian network instead of the decision tree.

References

[1] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997.

[2] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19:1300–1305, 1997.

[3] K. Cho and S. M. Dunn. Learning shape classes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(9):882–888, 1994.

[4] J. J. Ferri, P. Pudil, M. Hatef, and J. Kittler. Comparative study of techniques for large-scale feature selection. In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition in Practice IV*, pages 403–413. Elsevier Science B.V., 1994.

[5] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(9):891–906, 1991.

[6] E. J. Gibson and E. S. Spelke. The development of perception. In J. H. Flavell and E. M. Markman, editors, *Handbook of Child Psychology Vol. III: Cognitive Development*, chapter 1, pages 2–76. Wiley, 4th edition, 1983.

[7] B. W. Mel. Combining color, shape, and texture histogramming in a neurally-inspired approach to visual object recognition. *Neural Computation*, 9:777–804, 1997.

[8] H. Murase and S. K. Nayar. Visual learning and recognition of 3-D objects from appearance. *Int. J. Computer Vision*, 14:5–24, 1995.

[9] R. C. Nelson and A. Selinger. A cubist approach to object recognition. In *Int. Conf. on Computer Vision*, 1998.

[10] A. D. Pick. Improvement of visual and tactual form discrimination. *J. Exp. Psychol.*, 69:331–339, 1965.

[11] R. P. N. Rao and D. H. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence*, 78:461–505, 1995.

[12] B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. In *Fourth Europ. Conf. on Computer Vision*, Cambridge, UK, Apr. 1996.

[13] J. R. Silver and H. A. Rollins. The effects of visual and verbal feature-emphasis on form discrimination in preschool children. *J. Exp. Child Psychol.*, 16:205–216, 1973.

[14] M. Sur. MIT press release, 19 December, 1996.

[15] P. E. Utgoff and J. A. Clouse. A Kolmogorov-Smirnoff metric for decision tree induction. Computer Science Technical Report 96-3, University of Massachusetts, Amherst, 1996.

[16] T. Zelniker and L. Oppenheimer. Effect of different training methods on perceptual learning in impulsive children. *Child Development*, 47:492–497, 1976.