Toward True 3D Object Recognition

Jean Ponce[1], Svetlana Lazebnik[1], Fredrick Rothganger[1], Cordelia Schmid[2]
[1]Dept. of Computer Science and Beckman Institute, University of Illinois, Urbana, IL 61801, USA
[2]INRIA Rhône-Alpes, 665, Avenue de l'Europe, 38330 Montbonnot, France

Abstract: This paper addresses the problem of recognizing three-dimensional (3D) objects in photographs and image sequences. It revisits viewpoint invariants as a *local* representation of shape and appearance, and proposes a unified framework for object recognition where object models consist of a collection of small (planar) patches, their invariants, *and* a description of their 3D spatial relationship. This approach is applied to two fundamental instances of the 3D object recognition problem: (1) modeling rigid 3D objects from a small set of unregistered pictures and recognizing them in cluttered photographs taken from unconstrained viewpoints, and (2) recognizing non-uniform texture patterns despite appearance variations due to non-rigid transformations and changes in viewpoint. It is validated through several experiments, and extensions to the analysis of video sequences and the recognition of object categories are briefly discussed.

1 Introduction

We address the problem of recognizing 3D objects in photographs and image sequences. Today, the most popular approaches to object recognition are probably the *appearance-based* techniques [4, 10, 27, 46, 52], first proposed by Turk and Pentland [50] in the face recognition domain and by Murase and Nayar [31] in a more general context. They are directly related to classical methods from statistical pattern recognition [12]. In this framework, objects are typically represented by feature vectors, and the recognition problem is framed as one of supervised learning—training a classifier given a set of positive and negative examples. The problem explicitly addressed in this case is that of variability within a class. In most cases, issues arising from the systematic variation in the appearance of a 3D object due to varying viewpoint and illumination are dealt with implicitly if at all (see, however, [3, 31, 46]). In contrast, purely geometric approaches to 3D object recognition explicitly account for changes in viewpoint [17, 26, 48]. They are somewhat robust under changes in illumination since they typically discard the image brightness information in favor of binary features such as edges, but they are (mostly) limited to rigid objects observed in images with little or no clutter where segmentation is easy. Within-class variability is mostly addressed using structural object descriptions in terms or simple 3D geometric primitives [8, 33] with limited success in cluttered scenes with unconstrained viewpoint [55].

Viewpoint invariants (or *invariants* for short) provide a natural indexing mechanism for matching tasks as well as a bridge between appearance-based and geometric approaches to recognition. Unfortunately, although planar objects and certain simple shapes (e.g., bilaterally symmetric ones) admit invariants [32, 38], general 3D shapes do not [9], which is the main reason why invariants have fallen out of favor after an intense flurry of activity in the early 1990s [29, 30].

We propose to revisit invariants as a *local* representation of shape and appearance: Although smooth surfaces are almost never planar in the large, they are *always* planar in the small—that is, sufficiently small surface patches can always be thought of as being comprised of coplanar points. Thus, we propose a unified framework for object recognition where object models consist of a collection of small (planar) patches, their invariants, *and* a description of their 3D spatial relationship. Specifically, the local invariants used in our work are the affine-invariant descriptions of the image brightness pattern in the neighborhood of salient image features ("interest points" [15]) recently developed by Lindeberg and Gårding [24, 23] and by Mikolajczyk and Schmid [28]. These *affine-invariant patches* provide us with a *normalized* representation of the local object appearance, invariant under viewpoint and illumination changes, that can be used as a local measure of image, part, or object similarity. Depending on the recognition problem at hand, we propose different models of the spatial relationship between local invariants to represent the global object structure and drive the matching process. As shown in the rest of this presentation, these range from "hard" geometric consistency constraints in rigid object recognition tasks to "soft" models of the distribution of similar-looking patches in non-rigid texture classification tasks.

We apply the proposed framework to two concrete object recognition problems: (1) modeling rigid 3D objects from a small set of unregistered pictures and recognizing them in cluttered photographs taken from unconstrained viewpoints, and (2) recognizing non-uniform texture patterns despite appearance variations due to non-rigid transformations and changes in viewpoint. Our approach is validated through several experiments, and extensions to the analysis of video sequences and the recognition of object categories are briefly discussed. The interested reader is refered to [21, 22, 41] for more details.

2 Recognizing Rigid 3D Objects

We address in this section the problem of recognizing rigid 3D objects in photographs. We use the affine-invariant patches introduced by Lindeberg and Gårding [24, 23] and Mikolajczyk and Schmid [28] to represent local surface appearance and select promising matches between pairs of images or an object model and an image. We use geometric consistency constraints related to the multi-view geometry studied in the structure-from-motion literature [49] to represent the global object structure, retain correct matches, and discard incorrect ones. The experiments presented later in this section show that rigid object models can be acquired automatically from a few images, and effectively used in recognition tasks [41]. It would of course be interesting to generalize these results to the analysis of image sequences that contain articulated objects; we will come back to this issue in Section 4.

2.1 Affine-Invariant Patches

We use an implementation of the affine-invariant region detector proposed by Mikolajczyk and Schmid [28] to capture local appearance information. In this approach, the dependency of an image patch's appearance on affine transformations is eliminated by an iterative rectification process using (a) the second-moment matrix computed in the neighborhood of a point to normalize the shape of the corresponding image patch in an affine-invariant manner; (b) the local extrema of the normalized Laplacian over scale to determine the characteristic scale of the local brightness pattern; and (c) an affine-adapted Harris detector to determine the patch location. The output of the affine-invariant region detection/rectification process is a set of image patches in the shape of ellipses, together with the (affine) transformation mapping these ellipses onto a unit circle centered at the origin. This transformation is only defined up to a rotational ambiguity (this is intuitively obvious since a planar affine transformation is defined by six independent parameters but an ellipse is only defined by five parameters). We use image gradient information to eliminate this ambiguity. This allows us to turn the shape of an affine-invariant patch from an ellipse to a parallelogram, and to determine the six degrees of freedom of an affine *rectifying transformations* $\mathcal{R}$ that maps this corresponding parallelogram onto a square with unit edge half-length centered at the origin (Figure 1).

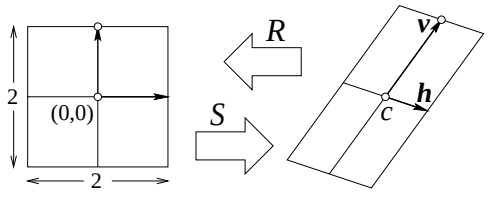


Figure 1: Geometric interpretation of the rectification matrix $\mathcal{R}$ and its inverse $\mathcal{S}$.

The rectified patch is a *normalized* representation of the local surface appearance that is invariant under planar affine transformations. We will assume from now on an affine—that is, orthographic, weak-perspective, or paraperspective—projection model. Under this model, our normalized appearance representation is invariant under arbitrary changes in viewpoint. For Lambertian patches and distant light sources, it can also be made invariant to changes in illumination (ignoring shadows) by subtracting the mean patch intensity from each pixel value and normalizing the sum of squared intensity values to one (or equivalently using *normalized* correlation to compare patches).

The rectifying transformation associated with a planar patch and its inverse can be represented by two 2×3 matrices $\mathcal{R}$ and $\mathcal{S}$ that map homogeneous (affine) plane coordinates onto non-homogeneous ones (Figure 1). These transformations play a fundamental role in the rest of this section. Let us first note that the columns vectors of the matrix $\mathcal{S}$ admit a simple geometric interpretation: Since they are respectively the images of the vectors $(1,0,0)^T$, $(0,1,0)^T$, and $(0,0,1)^T$ under that mapping, the third column $\boldsymbol{c}$ of $\mathcal{S}$ is the (non-homogeneous) coordinate vector of the patch center c, and its first two columns $\boldsymbol{h}$ and $\boldsymbol{v}$ are respectively the (non-homogeneous) coordinate vectors of the "horizontal" and "vertical" vectors joining c to the sides of the patch. The second key (and new) insight is that a rectified patch can also be thought of as a *fictitious* view of the original surface patch, and the inverse mapping $\mathcal{S}$ can thus be decomposed into an *inverse projection* $\mathcal{N}$ [13] that maps the rectified patch onto the corresponding surface patch, followed by a projection $\mathcal{M}$ that maps that patch onto its (true) image projection, i.e., $\mathcal{S} = \mathcal{M}\mathcal{N}$. Note that in the affine projection setting chosen here, we can write

$$\mathcal{M} = [\mathcal{A} \quad \boldsymbol{b}] \quad \text{and} \quad \mathcal{N} = \begin{bmatrix} \mathcal{B} \\ (0,0,1) \end{bmatrix},$$

where $\mathcal{A}$ and $\mathcal{B}$ are respectively 2×3 and 3×3 matrices, and $\boldsymbol{b}$ is a vector in $\mathbb{R}^2$.[3] The columns of the matrix $\mathcal{B}$ admit a geometric interpretation related to that of the matrix $\mathcal{S}$: Namely, the first two are the (non-homogeneous) coordinate vectors of the "horizontal" and "vertical" axes of the surface patch, and the third one is the (non-homogeneous) coordinate vector of its center C.

In particular (and not surprisingly), a match between $m \geq 2$ images of the same affine-invariant patches contains *exactly* the same information as a match between m triples of points. It is thus clear that all the machinery of structure from motion [49] and pose estimation [17, 26] from point matches can be exploited in modeling and object recognition tasks. Reasoning in terms of multi-view constraints associated with the matrix $\mathcal{S}$ provides a unified and convenient representation for all stages of both tasks.

2.2 Object Modeling

Let us assume for the time being that we are given n patches observed in m images, together with the corresponding 2×3 matrices $\mathcal{R}_{ij}$ and $\mathcal{S}_{ij}$ for $i = 1, \ldots, m$ and

[3] This is an affine instance of the characterization of homographies induced by planes given in Faugeras, Luong and Papadopoulo [13, Prop. 5.1].

$j = 1, \ldots, n$ (i and j serving respectively as image and patch indices). Following Tomasi and Kanade [49], we can take the center of mass of the observed patches' centers as the origin of the world coordinate system, and the center of mass of these points' projections as the origin of every image coordinate system. In this case, the vectors $\boldsymbol{b}_i$ are equal to zero, and we have $\mathcal{S}_{ij} = \mathcal{A}_i \mathcal{B}_j$, or equivalently, $\hat{\mathcal{S}} = \hat{\mathcal{A}}\hat{\mathcal{B}}$, where

$$\hat{\mathcal{S}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{S}_{11} & \ldots & \mathcal{S}_{1n} \\ \ldots & \ldots & \ldots \\ \mathcal{S}_{m1} & \ldots & \mathcal{S}_{mn} \end{bmatrix}, \hat{\mathcal{A}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{A}_1 \\ \vdots \\ \mathcal{A}_m \end{bmatrix}, \hat{\mathcal{B}} \stackrel{\text{def}}{=} [\, \mathcal{B}_1 \ \ldots \ \mathcal{B}_n \,].$$

In particular, $\hat{\mathcal{S}}$ has at most rank 3, a fact that can be used as a matching constraint when at least two matches are visible in at least two views. Alternatively, singular value decomposition can be used as in Tomasi and Kanade [49] to factor $\hat{\mathcal{S}}$ and compute estimates of the matrices $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$ that minimize the squared Frobenius norm of the matrix $\hat{\mathcal{S}} - \hat{\mathcal{A}}\hat{\mathcal{B}}$. Again, two views of two matches are sufficient to bring this constraint to bear on the matching process.

Image matching requires two key ingredients: (a) a measure of appearance similarity between two images of the same patch, and (b) a measure of geometric consistency between n matches $M_1, \ldots, M_n$ established across m images (a match is an m-tuple of image patches). For the former we use normalized correlation between rectified patches. For the latter, we use the method described in the previous section to estimate (when $m, n \geq 2$) the matrices $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$, and define $d(M_1, \ldots, M_n) = |\hat{\mathcal{S}} - \hat{\mathcal{A}}\hat{\mathcal{B}}|/\sqrt{3mn}$ as a measure of consistency among matches. In our current implementation, we only match patches across pairs of images ($m = 2$), and follow a strategy similar to that used in the range data domain by Johnson and Hebert [18] with *spin images*. Given a patch in one image, we first select its most promising matches in the second image based on normalized correlation of the rectified patches. We then discard the matches M such that the number of consistent matches M' (i.e., matches such that $d(M, M')$ is less than some preset threshold) is less than some fixed percentage of the total number of candidate matches. At this point, we find groups of consistent matches as follows: For each one of the surviving $p < n$ matches, we initialize the group G to that match M, we then find the match M' minimizing $d(G, M')$ (naturally defined as $d(M_1, \ldots, M_k, M')$ when $G = (M_1, \ldots, M_k)$). If $d(G, M')$ is smaller than a preset threshold, we add M' to G and continue. This results in the construction of p groups. Finally, we discard all groups smaller than a last threshold. The remaining matches are judged to be correct. The implementation of this matching strategy is determined by the choice of several thresholds; we will come back to this point later in this paper.

Results. The proposed matching strategy can be used in modeling tasks to match successive pairs of views of the same object. When some of the patches are only observed in some of the frames (the usual case), the data can be split into overlapping blocks of two or more frames, using all the patches visible in all images of the same block to run the factorization technique, then using the points common to overlapping blocks to register the successive reconstructions in a common frame. In principle, it is sufficient to have blocks that overlap by four points. Once all blocks are registered, the initial estimates of the variables $\mathcal{M}_i$ and $\mathcal{N}_j$ can be refined through a few non-linear least-squares iterations. When three or more views are available, it is then a simple matter to compute the corresponding Euclidean weak-perspective projection matrices (assuming the aspect-ratios are known) and recover the Euclidean structure of the scene [37]. Figure 2 shows the results of some modeling experiments. The modeling process is fully automatic, and 16 to 29 images of each object have been used to construct the six models shown in the figure.

Figure 2: Model gallery: sample input images and renderings of the corresponding models.

2.3 Object Recognition

Let us now assume that the method proposed in the previous section has been used to construct an object model consisting of affine-invariant patches and the corresponding $\mathcal{B}_j$ matrices. Let us also assume that $n \geq 2$ affine-invariant patches found in a test image have been matched to n patches from this model, and derive consistency constraints that must be satisfied by these matches. Let $\mathcal{S}_1$, $\ldots$, $\mathcal{S}_n$ denote the inverse rectification matrices associated with the corresponding patches in the test image. As before, we can always pick the center of mass of the n patch centers in the test image as the origin of its coordinate system, and change the origin of the world coordinate system so that it coincides with the center of mass of their

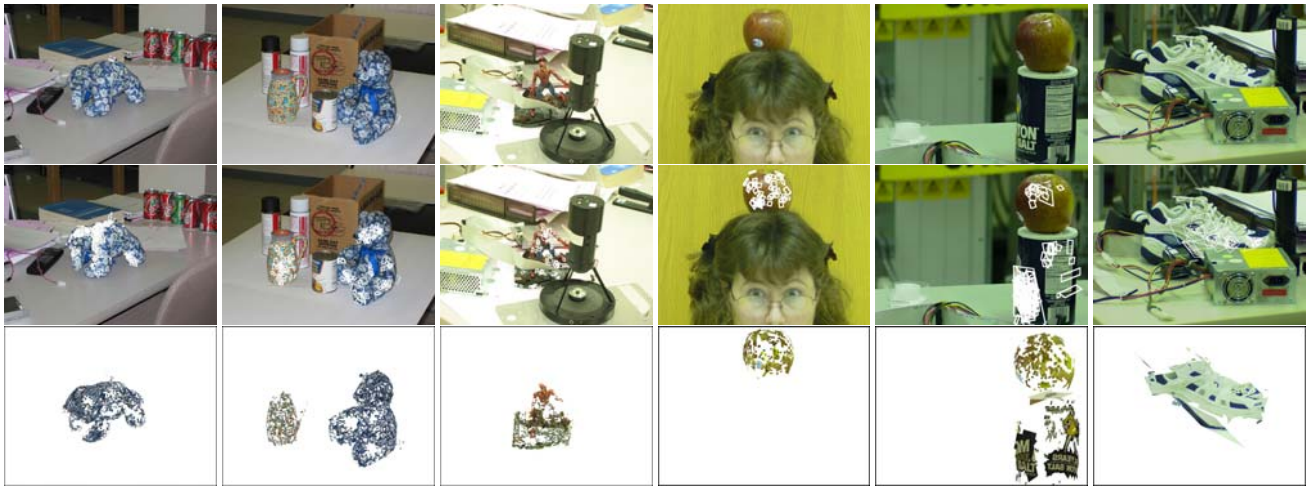


Figure 3: Object recognition experiments. The three rows of this figure show (respectively) input images, model patches matched to these images, and recognized models rendered in their estimated pose. Note that the teddy bear in the leftmost column is in a pose quite different from those used to acquire its model. Also note the significant amount of clutter and occlusion in each image.

matches in the model. With this convention, the projection matrix can be written as $\mathcal{M} = [\mathcal{A} \quad \mathbf{0}]$ and we can write as before $\mathcal{S}_j = \mathcal{A}\mathcal{B}_j$. We have therefore $\mathcal{A} = \mathcal{S}_j\mathcal{B}_j^{-1}$ for $j = 1, \ldots, p$. Note that the value of $\mathcal{B}_j$ is available from the modeling stage *in the coordinate system attached to the model*. If $\boldsymbol{C}$ denotes the (known) position of the center of mass of the patch centers in the original coordinate system, it is easy to see that the value of $\mathcal{B}_j$ in the new coordinate frame is obtained by subtracting $\boldsymbol{C}(0, 0, 1)$ from its old value. Given $n \geq 2$ patches, we write

$$\check{\mathcal{B}}\mathcal{A}^T = \check{\mathcal{S}}, \text{ where } \check{\mathcal{B}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{B}_1^T \\ \ldots \\ \mathcal{B}_n^T \end{bmatrix}, \text{ and } \check{\mathcal{S}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{S}_1^T \\ \ldots \\ \mathcal{S}_n^T \end{bmatrix},$$

which allows us to compute $\mathcal{A}^T$ as the solution of a linear least-squares problem. As in the image matching case, an appropriate measure of consistency is the normalized residual error $|\check{\mathcal{S}} - \check{\mathcal{B}}\mathcal{A}^T|/\sqrt{3n}$, that can once again be interpreted in terms of image distances.

Results. A matching strategy similar to the one used in object modeling can be used in object recognition tasks. In this case, we use the method described in the previous section to estimate (when $n \geq 2$) the matrix $\mathcal{A}^T$ and use $d'(M_1, \ldots, M_n) \stackrel{\text{def}}{=} |\check{\mathcal{S}} - \check{\mathcal{B}}\mathcal{A}^T|/3n$ as a measure of consistency among matches. Figure 3 shows several recognition results.

3 Recognizing Non-rigid Texture Classes

The strong geometric consistency constraints presented in Section 2 are appropriate for modeling and recognizing rigid and (as will be argued in Section 4) articulated objects. Here we take a first step toward recognizing object classes, where objects within the same class may not be related by any parametric transformation (think of two chairs, or a piece of cloth). In this context, it is still possible to represent objects locally by affine-invariant patches (non-rigid transformations are affine in the small, see the discussion below) and globally by the spatial relationship between these patches, but the geometric constraints involved have to be relaxed (see, for example, [1, 47, 53] for related work). We address in this section the (somewhat) simpler problem of representing and recognizing non-rigid textures observed from arbitrary viewpoints. We will come back to the general problem of category-level object recognition in Section 4, where we will propose using stronger spatial constraints (graphical descriptions of affine-invariant patch patterns) to represent the salient parts of objects.

Recent approaches to texture recognition [25, 36, 54] perform impressively well on datasets as challenging as the Brodatz database [7]. Unfortunately, these schemes rely on restrictive assumptions about their input (e.g., the texture must be stationary) and are not generally invariant under 2D similarity and affine transformations, much less 3D transformations caused by camera motions and non-rigid deformations of textured surfaces. In addition, most existing approaches to texture analysis use a dense representation where some local image descriptor is computed over a fixed neighborhood of each pixel. Affine-invariant patches can be used to address the issues of *spatial selection*—finding a sparse set of texture descriptors at "interesting" image locations—and *shape selection*—computing shape and scale characteristics of the descriptors—(see [44] for related work). In addition, they afford a texture representation that is invariant under any geometric transformation that can be *locally* approximated by an affine model: Local affine invariants are capable of modeling not only global affine transformations of the image, but also perspective distortions and non-rigid deformations that preserve the locally flat structure of the surface (e.g., the bending of paper or cloth). In this context, it is appropriate to combine the affine-invariant patches based on the Harris interest point detector and used in the previous section with the affine-adapted Laplacian blob detector proposed by Lindeberg and Gårding [24]. The two feature detectors are dubbed H

(for Harris) and L (for Laplacian) in the rest of this paper, and they provide two description "channels" for local image patterns. Their output on two sample images is shown in Figure 4. Intuitively, the two detectors provide complementary kinds of information: H responds to corners and other regions of "high information content" [28], while L produces a perceptually plausible decomposition of the image into a set of blob-like primitives.

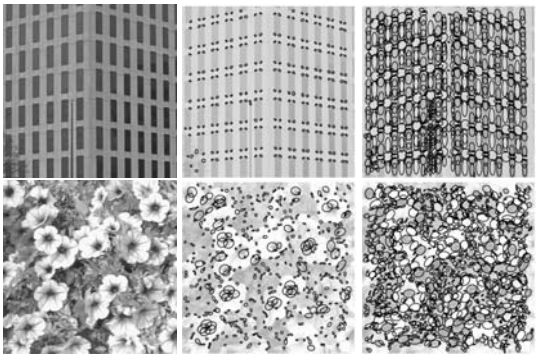


Figure 4: From left to right: building and flower images, H-detector output, L-detector output.

3.1 Intensity-Domain Spin Images

The affine-invariant patches found by the H and L region detectors can be thought of as the projections of ellipses drawn on the surface [24, 28]. Turning these ellipses into the parallelograms used in Section 2 is only possible when distinctive image gradient information is available. This is usually not the case for regions associated with extrema of the Laplacian since the image response is fairly uniform in these regions. In this case, the rectification process does not achieve a complete registration between two affinely transformed versions of the same texture patch, and a rotational ambiguity remains. This has motivated us to introduce a novel rotation-invariant descriptor of the local image brightness pattern, inspired by the *spin images* introduced by Johnson and Hebert [18] in the range data domain. The *intensity-domain spin image* is a two-dimensional histogram encoding the distribution of brightness values in an affine-normalized patch (Figure 5).

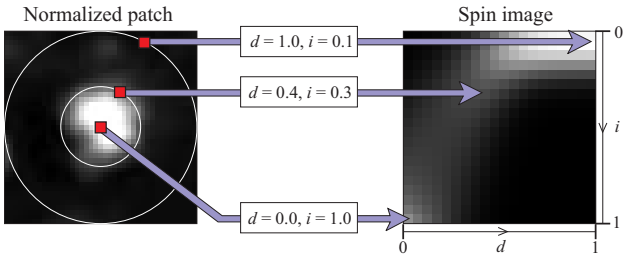


Figure 5: The construction of an intensity-domain spin image—three samples from a normalized patch are mapped onto their spin images.

The two dimensions of the histogram are d, the distance from the center or the origin of the normalized coordinate system of the patch, and i, the intensity value. The "slice" of the spin image corresponding to a fixed d is the histogram of the intensity values of pixels located at a distance d from the center. Since the d and i parameters are invariant to orthogonal transformations, intensity-domain spin images offer exactly the right degree of invariance for representing affine-normalized patches. To achieve invariance to affine transformations of the image intensity function (that is, transformations of the form $I \mapsto aI + b$), we use standard techniques to normalize the range of the intensity function within the support region of the spin image. The spin image is implemented as a "soft histogram", as advocated by Koenderink and Van Doorn [20], to reduce aliasing effects.

3.2 Images Signatures

This section demonstrates the power of affine-invariant patches and intensity-domain spin images as image descriptors in texture classification tasks. Our approach is illustrated by Figure 6, where the following process is applied to each image in the database using both the H and L feature detectors: (1) find the affine-invariant patches; (2) construct an affine-invariant description of these patches (the need for this step will be justified in the next section); and (3) find the most significant clusters of similar descriptions and use them to construct the *signature* [43] of the image. At the end of this process, all pairs of signatures are compared using the Earth Mover's Distance (EMD).

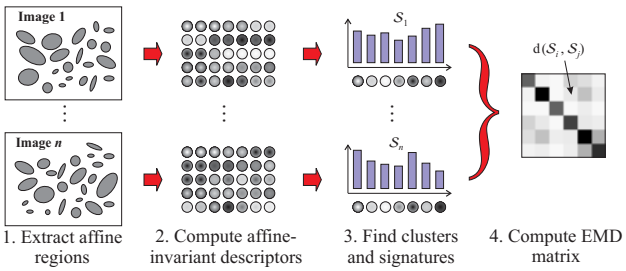
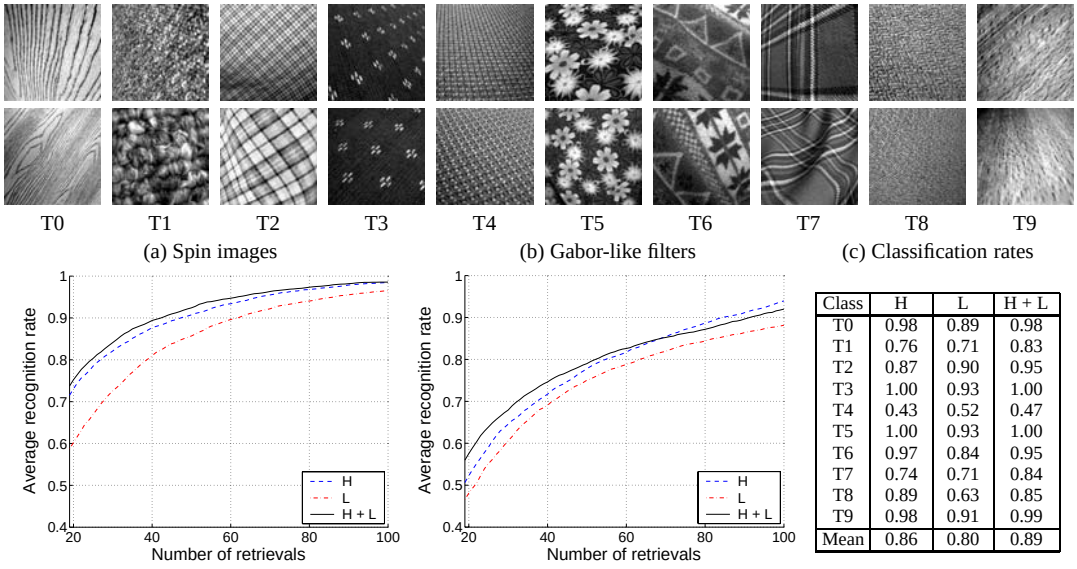


Figure 6: Architecture of the proposed texture recognition system.

Like most approaches to texture analysis, ours relies on clustering to discover a small set of basic primitives in the initial collection of candidate texture elements. We use a standard agglomerative clustering algorithm that iteratively merges clusters until either the desired target number of clusters is reached (10 to 15 in our implementation), or the distance between clusters exceeds a prespecified threshold. Agglomerative clustering takes as input not the descriptors (spin images) themselves, but only a *dissimilarity matrix* that records the distance between each pair of descriptors found in a particular image. After the clustering stage is completed, we form the final representation for the image: a *signature* of the form $\{(m_1, u_1), (m_2, u_2), \ldots, (m_k, u_k)\}$, where m_i is the *medoid* [19] (the most centrally located element of the ith cluster) and u_i is the relative weight of the cluster (the size of the cluster divided by the total number of descriptors extracted from the image). Signatures have been introduced by Rubner *et al.* [43] as representations suitable for matching using the *Earth Mover's Distance* (EMD). For our ap-



Class	H	L	H + L
T0	0.98	0.89	0.98
T1	0.76	0.71	0.83
T2	0.87	0.90	0.95
T3	1.00	0.93	1.00
T4	0.43	0.52	0.47
T5	1.00	0.93	1.00
T6	0.97	0.84	0.95
T7	0.74	0.71	0.84
T8	0.89	0.63	0.85
T9	0.98	0.91	0.99
Mean	0.86	0.80	0.89

Figure 7: Top: Samples of the ten texture classes used in our experiments. Bottom: Retrieval and classification results [22].

plication, the signature/EMD framework offers several important advantages. A signature is more descriptive than a histogram, and it does not require global clustering of the descriptors found in all images. In addition, EMD can match signatures of different sizes, and it is not very sensitive to the number of clusters—that is, if one cluster is split into several clusters with similar medoids, the magnitude of the EMD is not greatly affected. This is a very important property, since the automatic selection of the number of clusters remains a largely unsolved problem. Finally, recall that the proposed texture representation is designed to work with multiple channels corresponding to different affine-invariant region detectors (here, the H and L operators). Each channel generates its own signature representation for each image in the database, and therefore its own EMD value for any pair of images. We have experimented with several methods of combining the EMD matrices of the separate channels to arrive at a final estimate of the distance between each pair of images. Empirically, simply adding the distances produces the best results.

Results. We have implemented the proposed approach and conducted experiments with a dataset consisting of 200 images—20 samples each of ten different textured surfaces. Figure 7(top) shows three sample images of each texture. Significant viewpoint changes and scale differences are featured within each class. Several of the classes include additional sources of variability: inhomogeneities in the texture patterns, non-rigid transformations, illumination changes, and unmodeled viewpoint-dependent appearance changes. Figure 7(bottom, left) shows retrieval results using intensity-domain spin images as local image descriptors. Notice that for this dataset, the H channel is more discriminative than the L channel. Adding the EMD estimates provided by the two channels results in improved performance. Figure 7(bottom, center) shows the results obtained using the Gabor-like filters commonly used as image descriptors in texture analysis [45, 51] instead of intensity-domain spin images. Figure 7(bottom, right) summarizes the classification results obtained by using five samples from each class as training images. The classification rate for each class provides an indication of the "difficulty" of this class for our representation. The mean classification rate is 89% with two classes achieving 100%, showing the robustness of our system against a large amount of intra-class variability. Performance is very good for the rather inhomogeneous textures T5 and T6, but class T4 is not recognized very well, which is probably explained by the lack of an explicit model for viewpoint-dependent appearance changes caused by non-Lambertian reflectance and fine-scale 3D structure.

3.3 Generative Models

The previous section demonstrated the adequacy of our image descriptors in simple texture classification tasks. Here we go further and introduce generative models for the distribution of these descriptors, along with co-occurrence statistics for nearby patches. We use the EM algorithm to learn the generative model, which allows us to incorporate unsegmented multi-texture images into the training set. At recognition time, initial probabilities computed from the generative model are refined using a relaxation step that incorporates co-occurrence statistics learnt at modeling time. As mentioned above, the EM framework for learning texture models provides a natural way of incorporating unsegmented multi-texture images into the training set. Our approach is inspired by the work of Nigam et al. [34], who have proposed several techniques for using unlabeled data to improve the accuracy of text classification. Suppose we are given a multi-texture image annotated with the set $\mathcal{L}$ of class indices that it contains—that is, each feature vector $\boldsymbol{x}$ extracted from this image has an *incomplete* label of the form $C_{\mathcal{L}} = \{C_\ell | \ell \in \mathcal{L}\}$. We model the class-conditional density of the feature vectors $\boldsymbol{x}$ given the class labels ℓ

as $p(\boldsymbol{x}|C_\ell) = \sum_{m=1}^{M} p(\boldsymbol{x}|c_{\ell m})\, p(c_{\ell m})$, where the components $c_{\ell m}$, $m = 1, \ldots, M$, are thought of as *sub-classes*. Each $p(\boldsymbol{x}|c_{\ell m})$ is assumed to be a Gaussian with mean $\boldsymbol{\mu}_{\ell m}$ and covariance matrix $\Sigma_{\ell m}$. We estimate a single mixture model with $L \times M$ components using the EM algorithm to estimate the parameters of the model, including the mixing weights $p(c_{\ell m})$. We limit the number of free parameters in the optimization by using *spherical* Gaussians with covariance matrices of the form $\Sigma_{\ell m} = \sigma_{\ell m}^2 I$. This restriction also helps prevent the covariance matrices from becoming singular.

The estimation process starts by selecting some initial values for the parameters of the model (means, covariances, mixing weights). During the *expectation* or E-step, we use the parameters to compute probabilistic sub-class membership weights given the feature vectors $\boldsymbol{x}$ and the incomplete labels $C_\mathcal{L}$: $p(c_{\ell m}|\boldsymbol{x}, C_\mathcal{L}) \propto p(\boldsymbol{x}|c_{\ell m})\, p(c_{\ell m}|C_\mathcal{L})$, where $p(c_{\ell m}|C_\mathcal{L}) = 0$ for all $\ell \notin \mathcal{L}$ and $\sum_{\ell \in \mathcal{L}} \sum_{m=1}^{M} p(c_{\ell m}|C_\mathcal{L}) = 1$. During the *maximization* or M-step, we use the computed weights to re-estimate the parameters by maximizing the expected likelihood of the data in the standard fashion [5].

At this stage, each region in the training image is assigned the sub-class label that maximizes the posterior probability $p(c_{\ell m}|\boldsymbol{x}, C_\mathcal{L})$. Next, we need to a method for computing the neighborhood of a given region centered at pixel location $\boldsymbol{p}_0$ and described by the local shape matrix M. The simplest approach is to define the neighborhood as the set of all points $\boldsymbol{p}$ such that $(\boldsymbol{p} - \boldsymbol{p}_0)^T M (\boldsymbol{p} - \boldsymbol{p}_0) \leq \alpha$, where α is a constant factor. However, in practice this definition produces poor results: points with small ellipses get too few neighbors, and points with large ellipses get too many. A better approach is to "grow" the ellipse by adding a constant absolute amount (15 pixels in the implementation) to the major and minor axes, and to let the neighborhood consist of all points that fall inside this enlarged ellipse. In this way, the size and shape of the neighborhood still depends on the affine shape of the region, but the neighborhood structure is more balanced.

Once we have defined a neighborhood structure for the affine regions contained in an image, we can effectively turn this image into a directed graph with arcs emanating from the center of each region to other centers that fall within its neighborhood. The existence of an arc from a region with sub-class label c to another region with label c' is a joint event (c, c') (note that the order is important since the neighborhood relation is not symmetric). For each possible pair of labels, we estimate $p(c, c')$ from the relative frequency of its occurrence, and also find the marginal probabilities $\hat{p}(c) = \sum_{c'} p(c, c')$ and $\check{p}(c') = \sum_{c} p(c, c')$. Finally, we compute the values

$$r(c, c') = \frac{p(c, c') - \hat{p}(c)\, \check{p}(c')}{\left[\left(\hat{p}(c) - \hat{p}^2(c)\right)\left(\check{p}(c') - \check{p}^2(c')\right)\right]^{\frac{1}{2}}}$$

representing the correlations between the events that the labels c and c', respectively, belong to the source and destination nodes of the same arc. The values of $r(c, c')$ must lie between -1 and 1; negative values indicate that c and c' rarely co-occur as labels at endpoints of the same edge, while positive values indicate that they co-occur often.

In our experiments, we have found that the values of $r(c, c')$ are reliable only in cases when c and c' are sub-class labels of the same class C. Part of the difficulty in estimating correlations across texture classes is the lack of data in the training set. Even if the set contains multi-texture images, only a small number of edges actually fall across texture boundaries. Unless the number of texture classes is very small, it is also quite difficult to create a training set that would include samples of every possible boundary. Moreover, since we do not make use of "ground-truth" segmented images, the boundaries found in multi-texture images are not reliable. For these reasons, whenever c and c' belong to different classes, we set $r(c, c')$ to a constant negative value that serves as a "smoothness constraint" in the relaxation algorithm described next.

We have implemented the probability-based iterative relaxation algorithm described in the classic paper by Rosenfeld et al. [40] to enforce spatial consistency. The initial estimate of the probability that the ith region has label c, denoted $p_i^{(0)}(c)$, is obtained from the learned Gaussian mixture model as the posterior probability $p(c|\boldsymbol{x}_i)$. Note that since we run relaxation on unlabeled test data, these probabilities must be computed for all $L \times M$ sub-class labels corresponding to all possible classes. At each iteration, new probability estimates $p_i^{(t+1)}(c)$ are obtained by updating the current values $p_i^{(t)}(c)$ using the equation

$$p_i^{(t+1)}(c) = \frac{p_i^{(t)}(c)\left[1 + q_i^{(t)}(c)\right]}{\sum_c p_i^{(t)}(c)\left[1 + q_i^{(t)}(c)\right]},$$

$$q_i^{(t)}(c) = \sum_j w_{ij} \left[\sum_{c'} r(c, c')\, p_j^{(t)}(c')\right]. \qquad (1)$$

The scalars w_{ij} are weights that indicate how much influence region j exerts on region i. We treat w_{ij} as a binary indicator variable that is nonzero if and only if the jth region belongs to the ith neighborhood. Note that the weights are required to be normalized so that $\sum_j w_{ij} = 1$ [40].

It should be noted that the relaxation process iterating (1) has no convergence guarantees, though the constraints built into the update equation do ensure that the $p_i^{(t)}(c)$ stay non-negative and sum to 1 [40]. Despite the lack of a formal convergence proof, we have found the relaxation algorithm to behave well on our data. In practice, we run relaxation for 200 iterations.

Results. We have implemented the proposed approach (Figure 8). In our experiments, individual regions are classified in the obvious way, by assigning them to the class that maximizes $p_i(C_\ell) = \sum_{m=1}^{M} p_i(c_{\ell m})$. To perform classification and retrieval at the image level, we need to define a "global" score for each texture class. In our experiments, the score for class C_ℓ is computed by summing the

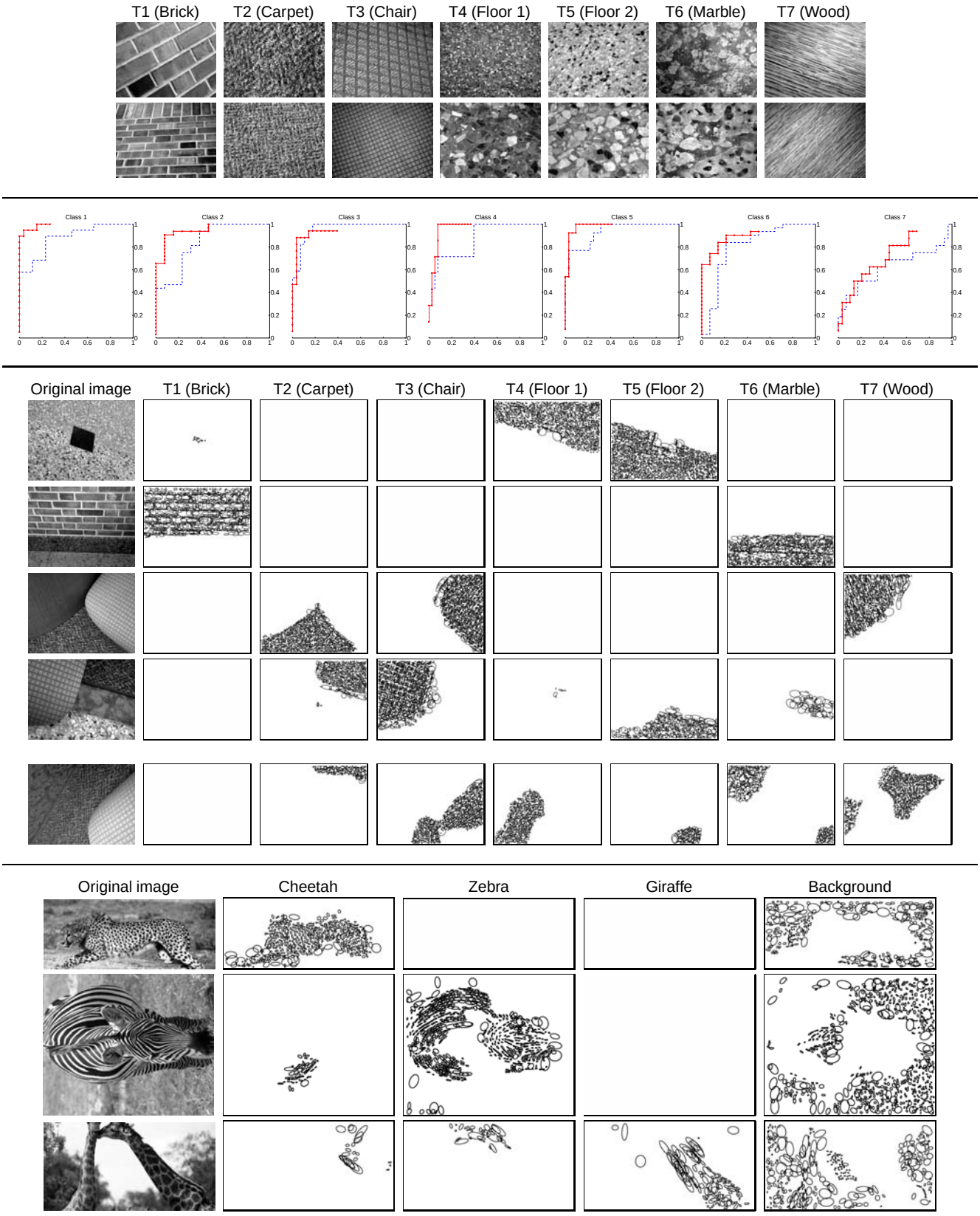


Figure 8: Segmentation/classification results. From top to bottom: Sample images of texture classes from an indoor scene; ROC curves (positive detection rate vs. false detection rate) for retrieval in the test set of 45 multi-texture images; four successful indoor image segmentation experiments (not that the two marble patches with different orientations are correctly classified in the second experiment), along with an unsuccessful one; animal image classification examples See [21] for additional results.

probability of C_ℓ over all N regions found in the image: $\sum_{i=1}^{N}\sum_{m=1}^{M} p_i(c_{\ell m})$, where the $p_i(c_{\ell m})$ are the probability estimates following relaxation. Classification of single-texture images is carried out by assigning the image to the class with the highest score, and retrieval for a given texture model proceeds from highest scores to lowest.

Our first data set contains seven different textures present in a single indoor scene. Figure 8(top) shows two sample images of each texture. The data set is partitioned as follows: 10 single-texture training images of each class; 10 single-texture validation images of each class; 13 two-texture training images; and 45 multi-texture test images. The next row of the figure show classification results in the form of ROC curves that plot the positive detection rate (the number of correct images retrieved over the total number of correct images) against the false detection rate (the number of false positives over the total number of negatives in the data set). Typical classification/segmentation results are shown next to illustrate the qualitative behavior of our algorithm. Our second data set consists of unsegmented images of three kinds of animals: cheetahs, giraffes, and zebras. The training set contains 10 images from each class, and the test set contains 20 images from each class, plus 20 "negative" images not containing instances of the target animal species. To account for the lack of segmentation, we introduce an additional "background" class, and each training image is labeled as containing the appropriate animal and the background. Typical classification examples are show in Figure 8(bottom). Overall, our system appears to learn very good models for cheetahs and zebras, but not for giraffes [21].

4 Going Further

We have presented a new framework for object recognition where object models consist of a collection of small (planar) patches, their invariants, and a description of their 3D spatial relationship. We believe that our experiments with 3D rigid object recognition and non-rigid texture recognition demonstrate the promise of this approach. To go further, we plan to attack two other fundamental object recognition problems: recognizing articulated objects in image sequences, with applications to the identification of shots that depict the same scene (*shot matching*) in video clips; and learning and recognizing part-based descriptions of 3D object classes in photographs and video clips.

Solving the first of these problems involves overcoming several challenges, including motion segmentation [6, 11, 14] in the difficult case where both the camera and parts of the scene may be moving independently, and matching 3D models acquired from different video clips. Figure 9 illustrates our preliminary efforts, with 3D models of a teddy bear extracted from two video sequences and matched with each other.

The second problem—learning and recognizing object class models from images—remains largely unsolved despite 40 years of efforts. Several recent approaches to category-level object recognition use machine learning techniques to acquire part models from training images, then train a classifier to recognize objects using the spatial layout of these parts in an image. This paradigm has been successfully applied to the recognition of cars [1, 47, 53], faces [16, 42, 47], and human beings [35, 39] in complex imagery. However, the image descriptors used in these methods enjoy very limited invariance properties (mostly translational invariance, see [1, 53] for example), which severely limits the range of admissible viewpoints that they can handle. We propose using graphical models of the characteristic patterns formed by affine-invariant patches to describe salient object parts. Figure 10 illustrates this idea with matches found between face images using the output of the L operators and a variant of affine alignment [2]. In these examples at least, the patch patterns are stable despite large viewpoint variations and appearance changes. We are in the process of assessing the true potential of this approach.

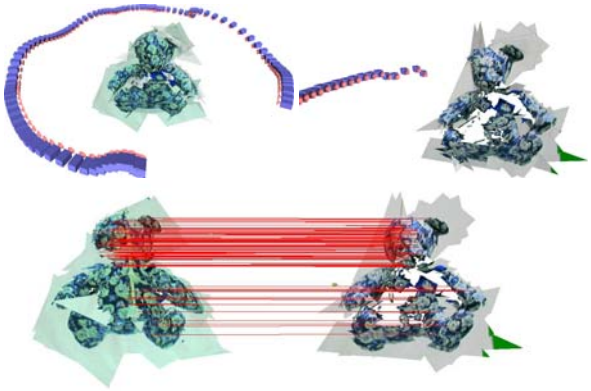


Figure 9: Modeling and recognizing objects in video sequences. Top: bear models constructed from two video clips, along with the recovered camera trajectories; Bottom: matching results.

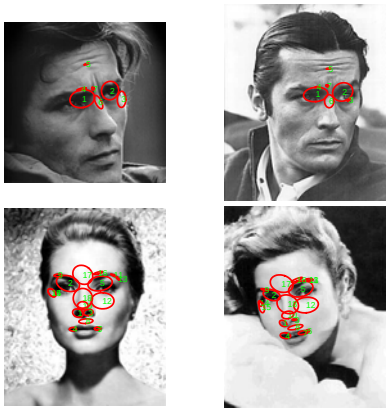


Figure 10: Matching faces.

Acknowledgments. This research was partially supported by the Beckman Institute, the UIUC Campus Research Board, the National Science Foundation under grants IIS-0308087 and IIS-0312438, the CNRS-UIUC Research Collaboration Agreements, and the European projects VIBES and LAVA.

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proc. European Conf. Comp. Vision*, volume LNCS 2353, pages 113–127, Copenhagen, Denmark, 2002.

[2] R. Basri and S. Ullman. The alignment of objects with smooth surfaces. In *Proc. Int. Conf. Comp. Vision*, pages 482–488, Tampa, FL, 1988.

[3] P.N. Belhumeur, J.P. Hesphanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class-specific linear projection. In *Proc. European Conf. Comp. Vision*, pages 45–58, 1996.

[4] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *Proc. Int. Conf. Comp. Vision*, pages 454–461, 2001.

[5] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[6] T.E. Boult and L.G. Brown. Factorization-based segmentation of motions. In *IEEE Workshop on Visual Motion*, pages 179–186, 1991.

[7] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, New York, 1966.

[8] R.A. Brooks. Symbolic reasoning among 3-D models and 2-D images. *Artificial Intelligence Journal*, 17(1-3):285–348, 1981.

[9] J. B. Burns, R. S. Weiss, and E. M. Riseman. View variation of point-set and line-segment features. *IEEE Trans. Patt. Anal. Mach. Intell.*, 15(1):51–68, January 1993.

[10] O. Carmichael and M. Hebert. Object recognition by a cascade of edge probes. In *British Machine Vision Conf.*, 2002.

[11] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *Proc. Int. Conf. Comp. Vision*, pages 1071–1076, Boston, MA, 1995.

[12] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-Interscience, 2001. Second edition.

[13] O. Faugeras, Q.-T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images*. MIT Press, 2001.

[14] C.W. Gear. Multibody grouping in moving objects. *Int. J. of Comp. Vision*, 29(2):133–150, August/September 1998.

[15] C. Harris and M. Stephens. A combined edge and corner detector. In 4^{th} *Alvey Vision Conference*, pages 189–192, Manchester, UK, 1988.

[16] B. Heisele, T. Serre, M. Pontil, and T. Poggio. Component-based face detection. In *Proceedings IEEE Conference Computer Vision and Pattern Recognition*, pages 657–662, 2001.

[17] D.P. Huttenlocher and S. Ullman. Object recognition using alignment. In *Proc. Int. Conf. Comp. Vision*, pages 102–111, London, U.K., June 1987.

[18] A.E. Johnson and M. Hebert. Surface matching for object recognition in complex three-dimensional scenes. *Image and Vision Computing*, 16:635–651, 1998.

[19] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data. An Introduction to Cluster Analysis*. John Wiley and sons, New York, 1990.

[20] J.J. Koenderink and A.J. Van Doorn. The structure of locally orderless images. *Int. J. of Comp. Vision*, 31(2/3):159–168, 1999.

[21] S. Lazebnik, C. Schmid, and J. Ponce. Affine-invariant local descriptors and neighborhood statistics for texture recognition. In *Proc. Int. Conf. Comp. Vision*, 2003.

[22] S. Lazebnik, C. Schmid, and J. Ponce. Sparse texture representations using affine-invariant neighborhoods. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2003.

[23] T. Lindeberg. Feature detection with automatic scale selection. *Int. J. of Comp. Vision*, 30(2):79–116, 1998.

[24] T. Lindeberg and J. Gårding. Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure. *Image and Vision Computing*, 15(6):415–434, 1997.

[25] F. Liu and W. Picard. Periodicity, directionality, and randomness: World features for image modeling and retrieval. *IEEE Trans. Patt. Anal. Mach. Intell.*, 18(7):722–733, 1996.

[26] D.G. Lowe. The viewpoint consistency constraint. *Int. J. of Comp. Vision*, 1(1):57–72, 1987.

[27] S. Mahamud, M. Hebert, and J. Lafferty. Combining simple discriminators for object discrimination. In *Proc. European Conf. Comp. Vision*, Copenhagen, Denmark, May 2002.

[28] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. European Conf. Comp. Vision*, volume I, pages 128–142, Copenhagen, Denmark, 2002.

[29] J.L. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision*. MIT Press, Cambridge, Mass., 1992.

[30] J.L. Mundy, A. Zisserman, and D. Forsyth. *Applications of Invariance in Computer Vision*, volume 825 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[31] H. Murase and S. Nayar. Visual learning and recognition of 3D objects from appearance. *Int. J. of Comp. Vision*, 14(1):5–24, 1995.

[32] V.S. Nalwa. Line-drawing interpretation: bilateral symmetry. In *Proc. DARPA Image Understanding Workshop*, pages 956–967, Los Angeles, CA, February 1987.

[33] R. Nevatia and T.O. Binford. Description and recognition of complex curved objects. *Artificial Intelligence Journal*, 8:77–98, 1977.

[34] K Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2–3):103–134, 2000.

[35] C.P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Proc. Int. Conf. Comp. Vision*, pages 555–562, 1998.

[36] R. Picard, T. Kabir, and F. Liu. Real-time recognition with the entire Brodatz texture database. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, pages 638–639, New York City, NY, 1993.

[37] C.J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19(3):206–218, March 1997.

[38] J. Ponce, D. Chelberg, and W. Mann. Invariant properties of straight homogeneous generalized cylinders and their contours. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11(9):951–966, September 1989.

[39] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *Proc. European Conf. Comp. Vision*, volume IV, pages 700–714, Copenhagen, Denmark, 2002.

[40] A. Rosenfeld, R. Hummel, and S. Zucker. Scene labeling by relaxation operations. *IEEE Trans. on Systems, Man, and Cybernetics*, 6(6):420–433, 1976.

[41] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, 2003.

[42] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. Patt. Anal. Mach. Intell.*, 20(1):23–38, 1998.

[43] Y. Rubner, C. Tomasi, and L.J. Guibas. A metric for distributions with applications to image databases. In *Proc. Int. Conf. Comp. Vision*, 1998.

[44] F. Schaffalitzky and A. Zisserman. Viewpoint invariant texture matching and wide baseline stereo. In *Proc. Int. Conf. Comp. Vision*, Vancouver, Canada, 2001.

[45] C. Schmid. Constructing models for content-based image retrieval. In *CVPR*, 2001.

[46] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19(5):530–535, May 1997.

[47] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *Proc. IEEE Conf. Comp. Vision Patt. Recog.*, Hilton Head, SC, 2000.

[48] A. Sethi, D. Renaudie, D.J. Kriegman, and J. Ponce. Curve and surface duals and the recognition of curved 3D objects from their silhouette. *Int. J. of Comp. Vision*, 2003. In press.

[49] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. of Comp. Vision*, 9(2):137–154, 1992.

[50] M. Turk and A.P. Pentland. Face recognition using eigenfaces. *J. of Cognitive Neuroscience*, 3(1), 1991.

[51] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *Proc. European Conf. Comp. Vision*, 2002.

[52] P. Viola and M. Jones. Robust real-time object detection. Technical Report CRL 01/01, Compaq Cambridge Research Laboratory, 2001.

[53] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proc. European Conf. Comp. Vision*, Dublin, Ireland, 2000.

[54] K. Xu, B. Georgescu, D. Comaniciu, and P. Meer. Performance analysis in content-based retrieval with textures. In *Proc. Int. Conf. Patt. Recog.*, 2000.

[55] M. Zerroug and R. Nevatia. From an intensity image to 3D segmented descriptions. In J. Ponce, A. Zisserman, and M. Hebert, editors, *Object Representation in Computer Vision II*, number 1144 in Lecture Notes in Computer Sciences, pages 11–24. Springer-Verlag, 1996.