3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints

Fred Rothganger, Svetlana Lazebnik, Cordelia Schmid, Jean Ponce

3D Object Modeling and Recognition Using Local Affine-Invariant Image Descriptors and Multi-View Spatial Constraints

Fred Rothganger (rothgang@uiuc.edu)
Svetlana Lazebnik (slazebni@uiuc.edu)
Department of Computer Science and Beckman Institute
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Cordelia Schmid (cordelia.schmid@inrialpes.fr)
INRIA Rhône-Alpes
665, Avenue de l'Europe, 38330 Montbonnot, France

Jean Ponce (jponce@uiuc.edu)
Department of Computer Science and Beckman Institute
University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA

Abstract. This article introduces a novel representation for three-dimensional (3D) objects in terms of local affine-invariant descriptors of their images and the spatial relationships between the corresponding surface patches. Geometric constraints associated with different views of the same patches under affine projection are combined with a normalized representation of their appearance to guide matching and reconstruction, allowing the acquisition of true 3D affine and Euclidean models from multiple unregistered images, as well as their recognition in photographs taken from arbitrary viewpoints. The proposed approach does not require a separate segmentation stage, and it is applicable to highly cluttered scenes. Modeling and recognition results are presented.

Keywords: Three-dimensional object recognition, image-based modeling, affine-invariant image descriptors, multi-view geometry.

1. Introduction

This article addresses the problem of recognizing three-dimensional (3D) objects in photographs. Traditional feature-based geometric approaches to this problem—such as alignment (Ayache and Faugeras, 1986; Faugeras and Hebert, 1986; Grimson and Lozano-Pérez, 1987; Huttenlocher and Ullman, 1987; Lowe, 1987) or geometric hashing (Thompson and Mundy, 1987; Lamdan and Wolfson, 1988; Lamdan and Wolfson, 1991)—enumerate various subsets of geometric image features before using pose consistency constraints to confirm or discard competing match hypotheses, but they largely ignore the rich source of information contained in the image brightness

and/or color pattern, and thus typically lack an effective mechanism for selecting promising matches. Appearance-based methods—as originally proposed in the context of face recognition (Turk and Pentland, 1991; Pentland et al., 1994; Belhumeur et al., 1997) and 3D object recognition (Murase and Nayar, 1995; Selinger and Nelson, 1999)—take the opposite view, and prefer to explicit geometric reasoning a classical pattern recognition framework (Duda et al., 2001) that exploits the discriminatory power of (relatively) low-dimensional, empirical models of global object appearance in classification tasks. However, they typically deemphasize the combinatorial aspects of the search involved in any matching task, which limits their ability to handle occlusion and clutter.

Viewpoint and/or illumination invariants (or *invariants* for short) provide a natural indexing mechanism for object recognition tasks. Unfortunately, although planar objects and certain simple shapes—such as bilateral symmetries (Nalwa, 1988) or various types of generalized cylinders (Ponce et al., 1989; Liu et al., 1993)—admit invariants, general 3D shapes do not (Burns et al., 1993), which is the main reason why invariants have fallen out of favor after an intense flurry of activity in the early 1990s (Mundy and Zisserman, 1992; Mundy et al., 1994). We propose in this article to revisit invariants as a *local* description of truly three-dimensional objects: Indeed, although smooth surfaces are almost never planar in the large, they are always planar in the small—that is, sufficiently small patches can be treated as being comprised of coplanar points.[1] The surface of a solid can thus be represented by a collection of small patches, their geometric and photometric invariants and a description of their 3D spatial relationships. The invariants provide an effective appearance filter for selecting promising match candidates in modeling and recognition tasks, and the spatial relationships afford efficient matching algorithms for discarding geometrically inconsistent candidate matches.

[1] Physical solids are of course not bounded by ideal smooth surfaces. We assume in the rest of this presentation that all objects of interest are observed from a relatively small range of distances, such that their surfaces appear geometrically smooth, and patches projecting onto small image regions are indeed roughly planar compared to the overall scene relief. This has proven reasonable in our experiments, where the apparent size of a given object never varies by a factor greater than five.

Concretely, we propose using local image descriptors that are invariant under affine transformations of the spatial domain (Gårding and Lindeberg, 1996; Lindeberg, 1998; Baumberg, 2000; Schaffalitzky and Zisserman, 2002; Mikolajczyk and Schmid, 2002) and of the brightness/color signal (Lowe, 2004) to capture the appearance of salient surface patches, and a set of multi-view geometric constraints related to those studied in the structure from motion literature (Tomasi and Kanade, 1992) to capture their spatial relationship. Our approach is directly related to a number of recent techniques that combine local models of image appearance in the neighborhood of salient features—or "interest points" (Harris and Stephens, 1988)—with local and/or global geometric constraints in wide-baseline stereo matching (Tell and Carlsson, 2000; Tuytelaars and Van Gool, 2004), image retrieval (Schmid and Mohr, 1997; Pope and Lowe, 2000), and object recognition tasks (Weber et al., 2000; Fergus et al., 2003; Mahamud and Hebert, 2003; Lowe, 2004). These methods normally either require storing a large number of views for each object (Schmid and Mohr, 1997; Pope and Lowe, 2000; Mahamud and Hebert, 2003; Lowe, 2004), or limiting the range of admissible viewpoints (Schneiderman and Kanade, 2000; Weber et al., 2000; Fergus et al., 2003). In contrast, our approach supports the automatic acquisition of explicit 3D affine and Euclidean object models from multiple unregistered images, and their recognition in heavily-cluttered pictures taken from arbitrary viewpoints.

The rest of this presentation is organized as follows: Section 2 presents the main elements of our approach. Its applications to 3D object modeling and recognition are discussed in Sections 3 and 4. In practice, object models are constructed in controlled situations with little or no clutter, and the stronger consistency constraints associated with 3D models make up for the presence of significant clutter and occlusion in recognition tasks, avoiding the need for a separate segmentation stage. Modeling and recognition examples can be found in Figures 1, 15–16, 20 and 26, and a detailed description of our experiments, including quantitative recognition results, can be found in Sections 3.3 and 4.5. We conclude in Section 5 with a brief discussion of the promise and limitations of the proposed approach.

Figure 1. Results of a recognition experiment. Left: A test image. Right: Instances of five models (a teddy bear, a doll stand, a salt can, a toy truck and a vase) have been recognized, and the models are rendered in the poses estimated by our program. Bounding boxes for the reprojections are shown as black rectangles.

A preliminary version of this article has appeared in (Rothganger et al., 2003).

2. Approach

This section presents the three main components of our approach to object modeling and recognition: (1) the *affine regions* that provide us with a normalized, viewpoint-independent description of local image appearance; (2) the geometric multi-view constraints associated with the corresponding surface patches; and (3) the algorithms that enforce both photometric and geometric consistency constraints while matching groups of affine regions in modeling and recognition tasks.

2.1. Affine Regions

The construction of local invariant models of object appearance involves two steps, the detection of salient image regions, and their description. Ideally, the regions found in two images of the same object should be the projections of the same surface patches. Therefore, they must be *covariant*, with regions detected in the first picture mapping onto those found in the second one via the geometric and photometric transformations induced by the corresponding viewpoint and illumination changes. In turn, detection

must be followed by a description stage that constructs a region representation *invariant* under these changes. For small patches of smooth Lambertian surfaces, the transformations are (to first order) affine, and this section presents the approach to detection and description of affine regions (Gårding and Lindeberg, 1996; Mikolajczyk and Schmid, 2002) used in our implementation.

2.1.1. *Detection*

Several approaches to finding perceptually-salient blob-like image primitives in natural images were proposed in the mid-eighties (Crowley and Parker, 1984; Voorhees and Poggio, 87). Blostein and Ahuja (1989) took a first step toward building some invariance in this process with a multi-scale region detector based on maxima of the Laplacian. Lindeberg (1998) has extended this detector in the framework of automatic scale selection, where a "blob" is defined by a scale-space location where a normalized Laplacian measure attains a local maximum. Gårding and Lindeberg (1996) have also proposed an *affine adaptation* process based on the second moment matrix for finding affine image blobs. Recently, Mikolajczyk and Schmid (2002) have combined these ideas into an integrated affine region detector.[2] Briefly, their algorithm iterates over steps where (1) an elliptical image region is deformed to maximize the isotropy of the corresponding brightness pattern (shape adaptation, see Gårding and Lindeberg, 1996); (2) its characteristic scale is determined as a local extremum of the normalized Laplacian in scale space (scale selection, see Lindeberg, 1998); and (3) the Harris (1988) operator is used to refine the position of the ellipse's center (localization, see Mikolajczyk and Schmid, 2002). The scale-invariant interest point detector proposed in (Mikolajczyk and Schmid, 2001) provides an initial guess for this procedure, and the elliptical region obtained at convergence can be shown to be covariant under affine transformations (see Gårding and Lindeberg, 1996; Lindeberg, 1998; Mikolajczyk and Schmid, 2002 for additional details).

[2] For related approaches to scale and affine region detection, see Baumberg (2000), Kadir and Brady (2001), Schaffalitzky and Zisserman (2002), Matas et al. (2002), Lowe (2004), Tuytelaars and Van Gool (2004).

The affine region detection process used in this article implements both this algorithm and a simple variant where a difference-of-Gaussians (DoG) operator (Crowley and Parker, 1984; Voorhees and Poggio, 87; Lowe, 2004) replaces the Harris interest point detector. Note that this operator tends to find corners and points where significant intensity changes occur, while the DoG detector is (in general) attracted to the centers of roughly uniform regions (blobs). Intuitively, the two operators provide complementary kinds of information: The Harris detector responds to regions of "high information content" (Mikolajczyk and Schmid, 2002), while the DoG detector produces a perceptually plausible decomposition of the image into a set of blob-like primitives. Figure 2 shows examples of the outputs of these two detectors.

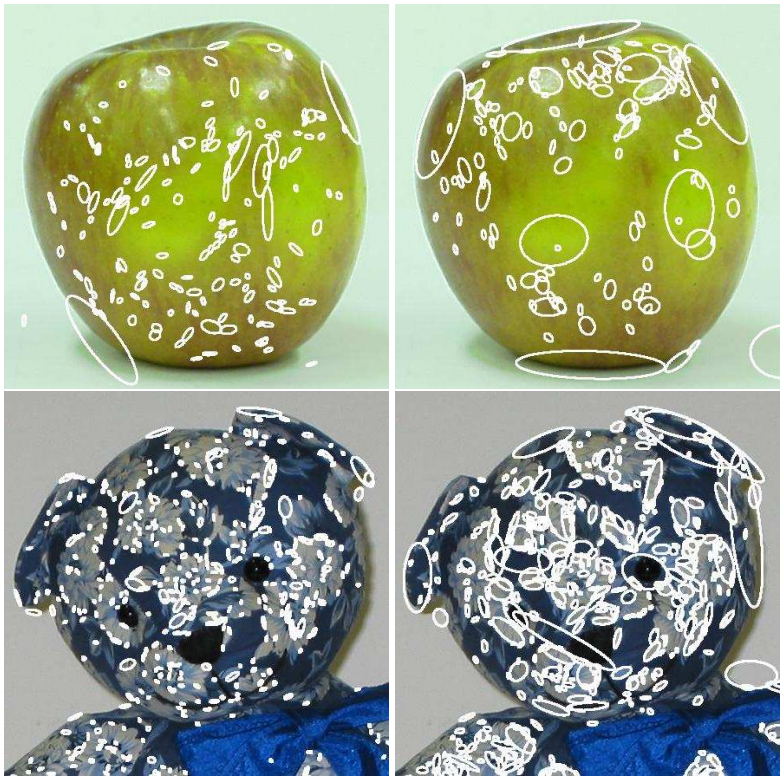


Figure 2. Affine-adapted patches found by Harris-Laplacian (left) and DoG (right) detectors.

2.1.2. *Description*

As mentioned above, the affine regions output by our detection process have an elliptical shape. It is easy to show that any ellipse can be mapped onto a unit circle centered at the origin using a one-parameter family of affine transformations separated from each other by arbitrary orthogonal transformations (intuitively, this follows from the fact that circles are unchanged by rotations and reflections about their centers). This ambiguity can be resolved by determining the dominant gradient orientation of the image region (Lowe, 2004), turning the corresponding ellipse into a parallelogram and the unit circle into a square (Figure 3). Thus, the output of the detection process is a set of image regions in the shape of parallelograms, together with affine *rectifying transformations* that map each parallelogram onto a "unit" square centered at the origin (Figure 4).

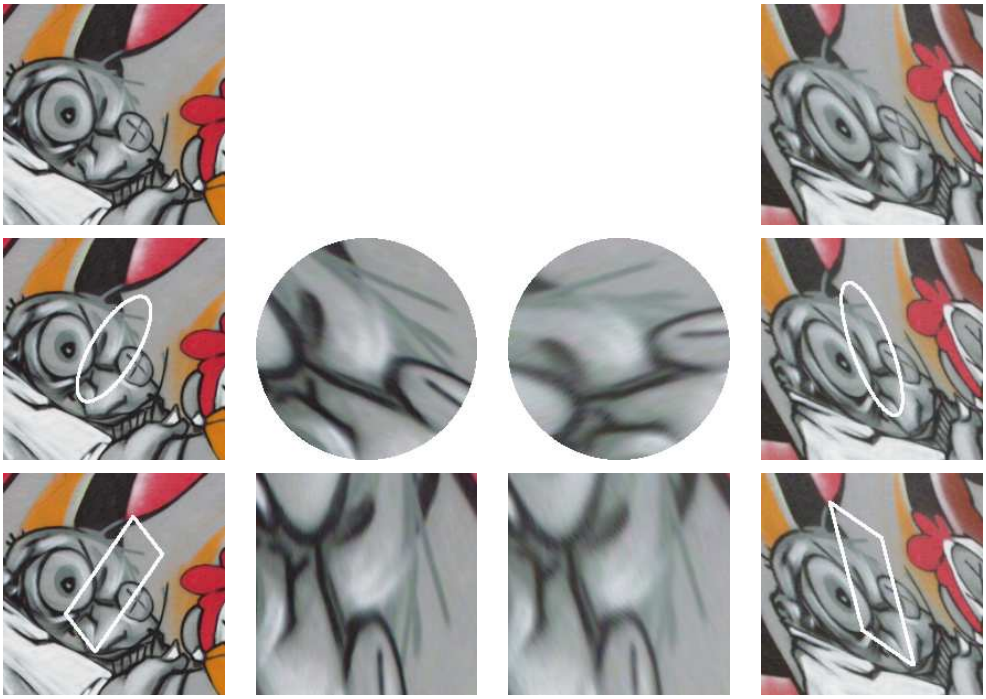


Figure 3. Normalizing patches. The left two columns show a patch from image 1 of Krystian Mikolajczyk's graffiti dataset (available from the INRIA LEAR Group's web page: `http://lear.inrialpes.fr/software`). The right two columns show the matching patch from image 4. The first row shows a portion of the original image. The second row shows the ellipse determined by affine adaptation. This normalizes the shape, but leaves a rotation ambiguity, as illustrated by the normalized circles in the center. The last row shows the same patches with orientation determined by the gradient at about twice the characteristic scale.

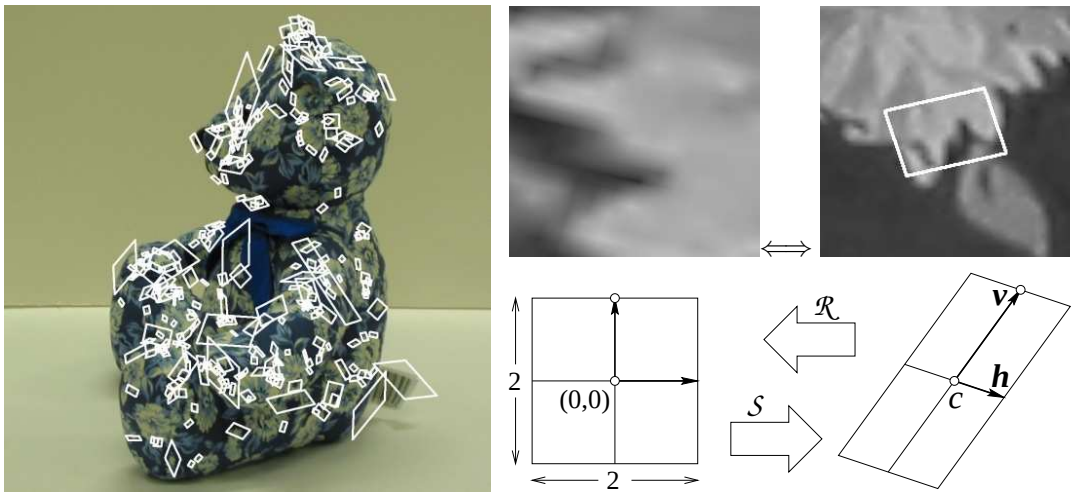


Figure 4. Affine regions. Left: A sample of the regions found in an image of a teddy bear (most of the patches actually detected in this image are omitted for clarity). Top right: A rectified patch and the original image region. Bottom right: Geometric interpretation of the rectification matrix $\mathcal{R}$ and its inverse $\mathcal{S}$ (see Section 2.2 for details).

A rectified affine region is a normalized representation of the local surface appearance, invariant under planar affine transformations. Under affine—that is, orthographic, weak-perspective, or para-perspective—projection models, this representation is invariant under arbitrary viewpoint changes. 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 Frobenius norm of the corresponding image array to one. Equivalently, normalized correlation can be used to compare rectified patches, irrespective of viewpoint and (affine) illumination changes. Maximizing correlation is equivalent to minimizing the squared distance between feature vectors formed by mapping every pixel value onto a separate vector coordinate. Other feature spaces may of course be used as well. In particular, the SIFT descriptor introduced by Lowe (2004) has been shown to provide superior performance in image retrieval tasks (Mikolajczyk and Schmid, 2003). Briefly, the SIFT description of an image region is a three-dimensional histogram over the spatial image dimensions and the gradient orientations, with the original rectangular area broken into 16 smaller ones, and the gradient directions

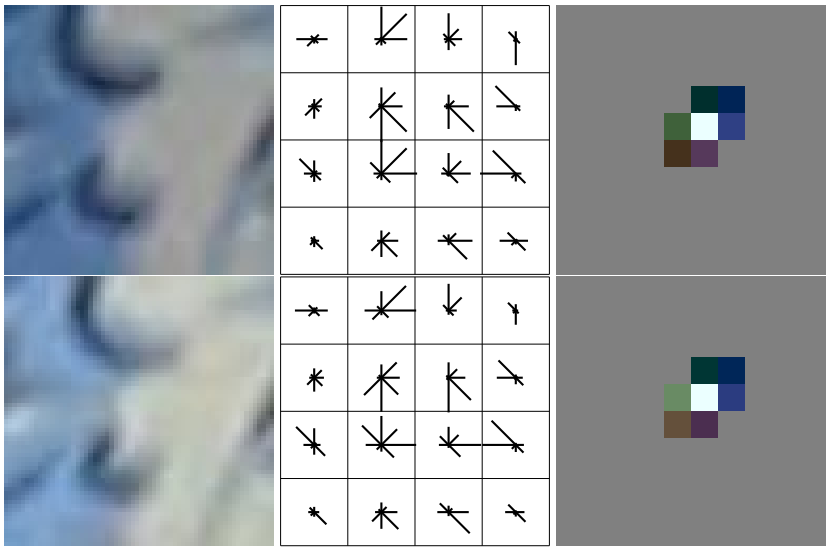


Figure 5. Two (rectified) matching patches found in two images of a teddy bear, along with the corresponding SIFT and color descriptors. Here (as in Figure 17 later), the orientation histogram values associated with each spatial bin are depicted by lines of different lengths for each one of the 8 quantized gradient orientations. As recommended in (Lowe, 2004), we scale the feature vectors associated with SIFT descriptors to unit norm, and compare them using the Euclidean distance. In this example, the distance is 0.28. The (monochrome) correlation of the two rectified patches is 0.9, and the χ^2 distance between the color histograms (as defined in Section 4.1) is 0.28. Each histogram appears as a grid of colored blocks, where the brightness of a block indicates the weight on that color. If a bin has zero weight, it appears as neutral gray.

quantized into 8 bins (Figure 5), and it can thus be represented by a 128-dimensional feature vector (Lowe, 2004).

In practice, our experiments have shown that combining the SIFT descriptor with a 10×10 color histogram drawn from the UV portion of YUV space improves the recognition rate in difficult cases with low-contrast patches. We will come back to this issue in Section 4.

2.2. Geometric Constraints

2.2.1. *Geometric Interpretation of the Rectification Process*

Let us denote by $\mathcal{R}$ and $\mathcal{S} = \mathcal{R}^{-1}$ the rectifying transformation associated with an affine region and its inverse. The 3×3 matrix $\mathcal{S}$ enjoys a simple geometric interpretation, illustrated by Figure 4 (bottom right), that will prove extremely useful in the sequel. It has the form

$$\mathcal{S} = \begin{bmatrix} \boldsymbol{h} & \boldsymbol{v} & \boldsymbol{c} \\ 0 & 0 & 1 \end{bmatrix}.$$

The matrix $\mathcal{R}$ is an affine transformation from the image patch to its rectified form, and thus $\mathcal{S}$ is an affine transformation from the rectified form back to the image patch. Since the center of the rectified patch has homogeneous coordinates $[0, 0, 1]^T$, the third column of $\mathcal{S}$ gives the homogeneous coordinates of the center c of the corresponding image parallelogram. Likewise, it is easy to see that $\boldsymbol{h}$ and $\boldsymbol{v}$ are the vectors joining c to the mid-points of the parallelogram's sides (Figure 4).

The matrix $\mathcal{S}$ effectively contains the locations of three points in the image, so a match between $m \geq 2$ images of the same patch 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 (Tomasi and Kanade, 1992) and pose estimation (Huttenlocher and Ullman, 1987; Lowe, 1987) from point matches can be exploited in modeling and object recognition tasks. Reasoning in terms of multi-view constraints associated with the matrix $\mathcal{S}$ will provide in the next section a unified and convenient representation for all stages of both tasks, but one should always keep in mind the simple geometric interpretation of the matrix $\mathcal{S}$ and the deeply rooted relationship between these constraints and those used in motion analysis and pose estimation.

2.2.2. *Multi-View Constraints*

Let us assume for the time being that we are given n patches observed in m images, together with the (inverse) rectifying transformations $\mathcal{S}_{ij}$ defined as in the previous section for $i = 1, \ldots, m$ and $j = 1, \ldots, n$ (i and j serving respectively as image and patch indices). We use these matrices to derive in this section a set of geometric and algebraic constraints that must be satisfied by matching image regions.

A rectified patch can be thought of as a fictitious view of the original surface patch (Figure 6), and the mapping $\mathcal{S}_{ij}$ can thus be decomposed into an *inverse projection* $\mathcal{N}_j$ (Faugeras et al., 2001) that maps the rectified patch onto the corresponding surface patch, followed by a projection $\mathcal{M}_i$ that maps that patch onto its projection in image number i. In particular, we can write $\mathcal{S}_{ij} = \mathcal{M}_i\mathcal{N}_j$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$,

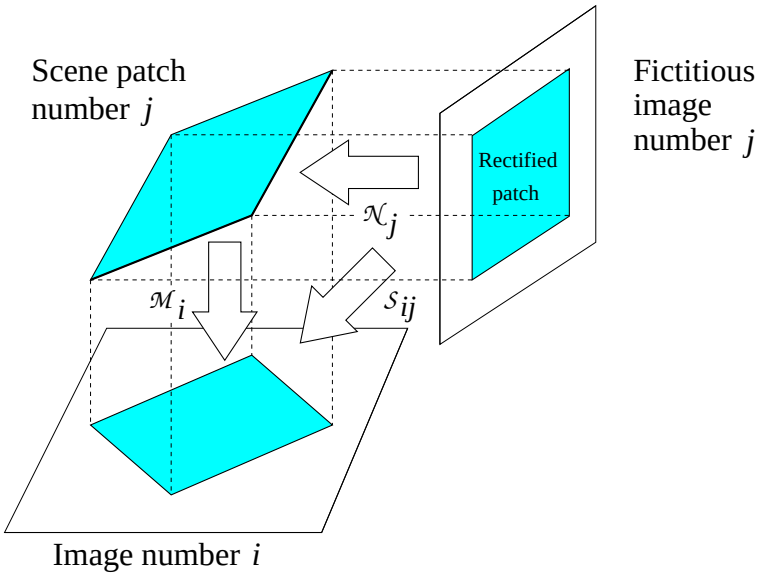


Figure 6. Geometric interpretation of the decomposition of the mapping $\mathcal{S}_{ij}$ into the product of a projection matrix $\mathcal{M}_i$ and an inverse projection matrix $\mathcal{N}_j$.

or, in a more compact form:

$$\hat{\mathcal{S}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{S}_{11} & \dots & \mathcal{S}_{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{S}_{m1} & \dots & \mathcal{S}_{mn} \end{bmatrix} = \begin{bmatrix} \mathcal{M}_1 \\ \vdots \\ \mathcal{M}_m \end{bmatrix} [\, \mathcal{N}_1 \quad \dots \quad \mathcal{N}_n \,] ,$$

and it follows that the $3m \times 3n$ matrix $\hat{\mathcal{S}}$ has at most rank 4.

As shown in Appendix A, the inverse projection matrix can be written as

$$\mathcal{N}_j = \begin{bmatrix} \boldsymbol{H}_j & \boldsymbol{V}_j & \boldsymbol{C}_j \\ 0 & 0 & 1 \end{bmatrix},$$

and it satisfies the constraint $\mathcal{N}_j^T \mathbf{\Pi}_j = \mathbf{0}$, where $\mathbf{\Pi}_j$ is the coordinate vector of the plane Π_j that contains the patch. In addition, the columns of the matrix $\mathcal{N}_j$ admit in our case a geometric interpretation related to that of the matrix $\mathcal{S}_{ij}$: Namely, the first two contain the "horizontal" and "vertical" axes of the surface patch, and the third one is the homogeneous coordinate vector of its center.

To account for the form of $\mathcal{N}_j$, we construct a reduced factorization of $\hat{\mathcal{S}}$ by picking, as in (Tomasi and Kanade, 1992), 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

projection equation $\mathcal{S}_{ij} = \mathcal{M}_i \mathcal{N}_j$ becomes

$$\begin{bmatrix} \mathcal{D}_{ij} \\ 0\ 0\ 1 \end{bmatrix} = \begin{bmatrix} \mathcal{A}_i & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathcal{B}_j \\ 0\ 0\ 1 \end{bmatrix}, \quad \text{or} \quad \mathcal{D}_{ij} = \mathcal{A}_i \mathcal{B}_j,$$

where $\mathcal{A}_i$ is a 2×3 matrix, $\mathcal{D}_{ij} = [\boldsymbol{h}_{ij}\ \boldsymbol{v}_{ij}\ \boldsymbol{c}_{ij}]$ is a 2×3 matrix, and $\mathcal{B}_j = [\boldsymbol{H}_j\ \boldsymbol{V}_j\ \boldsymbol{C}_j]$ is a 3×3 matrix. It follows that the reduced $2m \times 3n$ matrix

$$\hat{\mathcal{D}} = \hat{\mathcal{A}}\hat{\mathcal{B}}, \text{ where } \hat{\mathcal{D}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{D}_{11} & \dots & \mathcal{D}_{1n} \\ \vdots & \ddots & \vdots \\ \mathcal{D}_{m1} & \dots & \mathcal{D}_{mn} \end{bmatrix}, \ \hat{\mathcal{A}} \stackrel{\text{def}}{=} \begin{bmatrix} \mathcal{A}_1 \\ \vdots \\ \mathcal{A}_m \end{bmatrix}, \ \hat{\mathcal{B}} \stackrel{\text{def}}{=} [\, \mathcal{B}_1 \quad \dots \quad \mathcal{B}_n \,], \tag{1}$$

has at most rank 3.

2.2.3. *Matching Constraints*

The rank deficiency of the matrix $\hat{\mathcal{D}}$ can be used as a geometric consistency constraint when at least two potential matches are visible in at least two views. Alternatively, singular value decomposition can be used, as in (Tomasi and Kanade, 1992), to factorize $\hat{\mathcal{D}}$ and compute estimates of the matrices $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$ that minimize the squared Frobenius norm of the matrix $\hat{\mathcal{D}} - \hat{\mathcal{A}}\hat{\mathcal{B}}$. Geometrically, the (normalized) Frobenius norm $d = |\hat{\mathcal{D}} - \hat{\mathcal{A}}\hat{\mathcal{B}}|/\sqrt{3mn}$ of the residual can be interpreted as the root-mean-squared distance (in pixels) between the center and normalized side points of the patches observed in the image and those predicted from the recovered matrices $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$. Given n matches established across m images (a match is an m-tuple of image patches), the residual error d can thus be used as a measure of inconsistency between the matches.

Together with the normalized models of local shape and appearance proposed in Section 2.1.2, this measure will prove an essential ingredient of the approach to (pairwise) image matching presented in the next section. It will also prove useful in modeling tasks where the projection matrices are known but the 3D configuration $\mathcal{B}$ of a single patch is unknown, and in recognition tasks when the patches' configurations are known but a single projection matrix $\mathcal{A}$ is unknown. In general, Eq. (1) provides an over-constrained set of linear equations on the unknown parameters of the matrix $\mathcal{B}$ ($\hat{\mathcal{B}}$ with $n = 1$) in the former case, and an over-constrained set of linear constraints

on the unknown parameters of the matrix $\mathcal{A}$ ($\hat{\mathcal{A}}$ with $m = 1$) in the latter one. Both are easily solved using linear least-squares, and they determine the corresponding value of the residual error.

2.3. MATCHING

The core computational components of model acquisition and object recognition are matching procedures: In image-based modeling, we seek groups of matches between the affine regions found in two pictures that are consistent with both the local appearance models introduced in Section 2.1.2 and the geometric constraints expressed by Eq. (1). In object recognition, one image is replaced by an object model consisting of a collection of 3D patches, but the matching task and the underlying constraints are essentially the same. Both tasks can be understood in the *constrained-search* model proposed by Grimson (1990), who has shown that finding an optimal solution—maximizing, say, the number of matches such that photometric and geometric discrepancies are bounded by some threshold, or some other reasonable criterion—is in general intractable (i.e., exponential in the number of matched features) in the presence of uncertainty, clutter, and occlusion.

Various approaches to finding a reasonable set of geometrically-consistent matches have been proposed in the past, including *interpretation tree* (or *alignment*) techniques (Ayache and Faugeras, 1986; Faugeras and Hebert, 1986; Grimson and Lozano-Pérez, 1987; Huttenlocher and Ullman, 1987; Lowe, 1987), and *geometric hashing* (Lamdan and Wolfson, 1988; Lamdan and Wolfson, 1991). An alternative is offered by *robust estimation* algorithms, such as *RANSAC* (Fischler and Bolles, 1981), and its variants (Torr and Zisserman, 2000), and *median least-squares*, that consider candidate correspondences consistent with a small set of *seed* matches as *inliers* to be retained in a fitting process, while matches exceeding some inconsistency threshold are considered as *outliers* and rejected. Although, like all other heuristic approaches to constrained search, RANSAC and its variants are not guaranteed to output an optimal set of matches, they often offer a good compromise between the number of

feature combinations that have to be examined and the pruning capabilities afforded by appearance- and geometry-based constraints: In particular, the number of samples necessary to achieve a desired performance with high probability can easily be computed from estimates of the percentage of inliers in the dataset, and it is independent of the actual size of the dataset (Fischler and Bolles, 1981).

Briefly, RANSAC iterates over two steps: In the *sampling* stage, a (usually, but not always) minimal set of seed matches is chosen randomly, and it is used to estimate the geometric parameters of the fitting problem at hand. The *consensus* stage then adds to the initial seed all the candidate matches that are consistent with the estimated geometry. The process iterates until a sufficiently large consensus set is found, and the geometric parameters are finally re-estimated. Despite its attractive features, pure RANSAC only achieves moderate performance in the challenging object recognition experiments presented in Section 4, where clutter may contribute 90% or more of the detected regions. As will be shown in that section, the simple variant outlined in Algorithm 1 below achieves better results. This algorithm uses the idea of consensus from RANSAC while it seeks the maximal set of consistent matches between two sets of patches. It operates in three key steps, explained below.

Step 1 of the algorithm takes advantage of appearance constraints to reduce the practical cost of the search. It focuses the matching process on the portion of the space of all matches ($A \times B$) which is *a priori* most likely to be correct. Here we are using appearance similarity as a heuristic, since it cannot be a perfect indicator of correct matches. Noise present in actual image measurements lowers the appearance scores for some true matches. Furthermore, nothing prevents incorrect matches from appearing the same.

Step 2 applies RANSAC to the limited set of match hypotheses to find a geometrically consistent subset. Our assumption is that the largest such consistent set will contain mostly true matches. This establishes the geometric relationship between the two sets of patches. Proceeding to Step 3 is optional but useful, since it maximizes the number of resulting matches.

Input: Two sets of patches A and B.
Output: A set $T \subseteq A \times B$ of trusted matches.

Step 1: Appearance-based selection of potential matches.
• Initialize the set of putative matches P by finding patch pairs from $A \times B$ with high appearance similarity.
Step 2: Robust estimation.
• Apply robust estimation to find a set $T \subseteq P$ of geometrically consistent ("trusted") matches.
• Use consistency constraints to remove outliers from T.
Step 3: Geometry-based addition of matches.
repeat
  repeat
    • Form a geometric model r from T.
    • Replace T with all matches in P that are consistent with r.
  until T stops changing.
  • Use consistency constraints to remove outliers from T.
  • Re-estimate r from T.
  • Add more putative matches to P using r as a guide.
until P stops changing.

Algorithm 1: Overall Matching Procedure.

Step 3 explores the remainder of the space of all matches, seeking other matches which are consistent with the established geometric relationship between the two sets of patches. Obtaining a (nearly) maximal set of matches is useful for recognition (where the number of matches acts as a confidence measure) and for modeling (where they provide more coverage of the object).

The same overall matching procedure is used in both our modeling and recognition experiments. In practice, object models are constructed in controlled situations with little or no clutter. Algorithm 1 has proven extremely reliable in this case, irrespective of the RANSAC variant used in its second step (Section 3). The heavily cluttered images used in our recognition experiments are much more challenging, with different variants giving significantly different performances. An extensive experimental comparison between several reasonable choices is presented in Section 4.

3. 3D Object Modeling from Images

This section presents our approach to the automated acquisition of affine and Euclidean 3D object models from collections of unregistered photographs. These models consist of collections of 3D surface patches in the shape of parallelograms, along with the corresponding appearance models, defined in terms of the corresponding texture patterns and rectifying transformations. We will use the teddy bear shown in Figure 7 to illustrate some of the steps of the modeling process. Additional modeling experiments will be presented in Section 3.3.

3.1. Constructing Partial Models from Image Pairs

As shown in Section 2.2, two images of two surface patches are sufficient to estimate the corresponding (affine) projection matrices and 3D patch configurations. Thus, object models can be constructed by matching pairs of overlapping images—a process akin to wide-baseline stereo (Baumberg, 2000; Matas et al., 2002; Mikolajczyk and Schmid, 2002; Pritchett and Zisserman, 1998; Schaffalitzky and Zisserman, 2002; Tell and Carlsson, 2000; Tuytelaars and Van Gool, 2004) and (robust) structure from motion (Tomasi and Kanade, 1992; Weinshall and Tomasi, 1995; Poelman and Kanade, 1997)—before stitching the corresponding partial models into a complete one. While it is possible to select these pairs automatically (Schaffalitzky and Zisserman, 2002), we have chosen to specify them manually using prior knowledge of the modeling setup: Typically, we acquire a number of views roughly located in an equatorial ring around the modeled object, as well as a couple of top and/or bottom views. Accordingly, we match pairs of successive equatorial images, plus some additional pairs where a top or bottom view has enough overlap with one of those from the ring.

After processing through point detectors and affine adaptation, an image can be viewed as simply a collection of affine regions. For each pair of images, we apply Algorithm 1 to match the two sets of regions. The remainder of this section gives implementation specifics for the algorithm in the context of image matching.

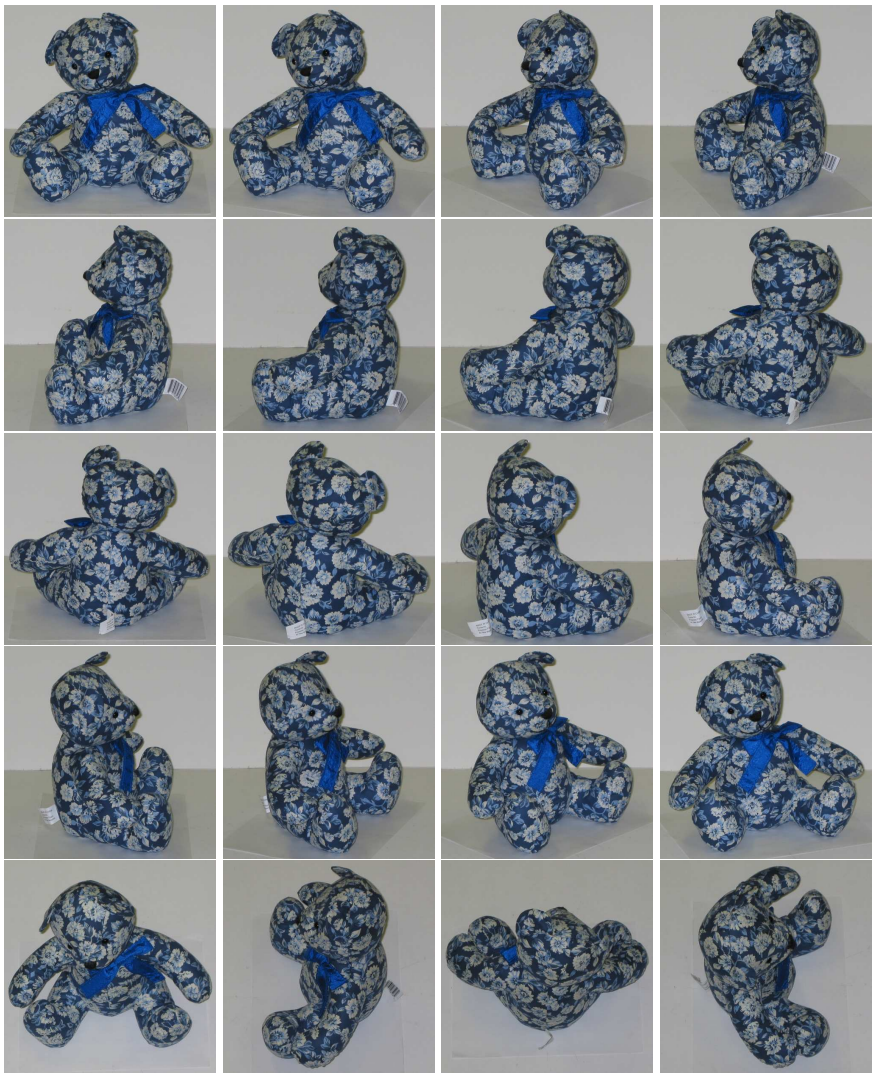


Figure 7. The 20 images used to construct the teddy bear model. There are 16 images roughly located in an equatorial ring, and 4 overhead images. This setup (with some variation in the number of input images) is typical of our modeling experiments.

3.1.1. *Appearance-Based Selection of Potential Matches*

We do not use color information in modeling tasks, and rely exclusively on SIFT feature vectors to characterize local image appearance. A *match* is an ordered pair of patches, one from the first image and one from the second image. The initial list of potential matches is found by selecting for each patch in the first image the top K patches in the second image as ranked by SIFT. In our experiments, K is typically set to 5, which is sufficient to model any of the objects. For objects with less distinctive

texture (specifically the apple and truck shown in Figure 16) it is useful to increase K to 10, which gives a richer set of matches. The cost of our (naive) implementation is $O(n^2 \log n)$, where n is the number of affine regions found in the two images. Using efficient (and possibly approximate) algorithms for finding the K nearest neighbors of a feature vector would obviously lower this cost, but this turns out to be negligible compared to the overall cost of Algorithm 1.

Candidate matches whose SIFT feature vectors are separated by a Euclidean distance greater than 0.5 are rejected. The remaining ones are used in the sampling stage of the matching procedure to estimate the projection matrices and seed its consensus step. For that process to be reliable, matching rectified regions should line up as well as possible despite the unavoidable imperfections of affine adaptation in real images. It is therefore desirable to adjust the parameters of one of the rectified regions to maximize correlation with its match. Appendix B presents a simple non-linear least-squares solution to this problem (see Figure 8 for an example).

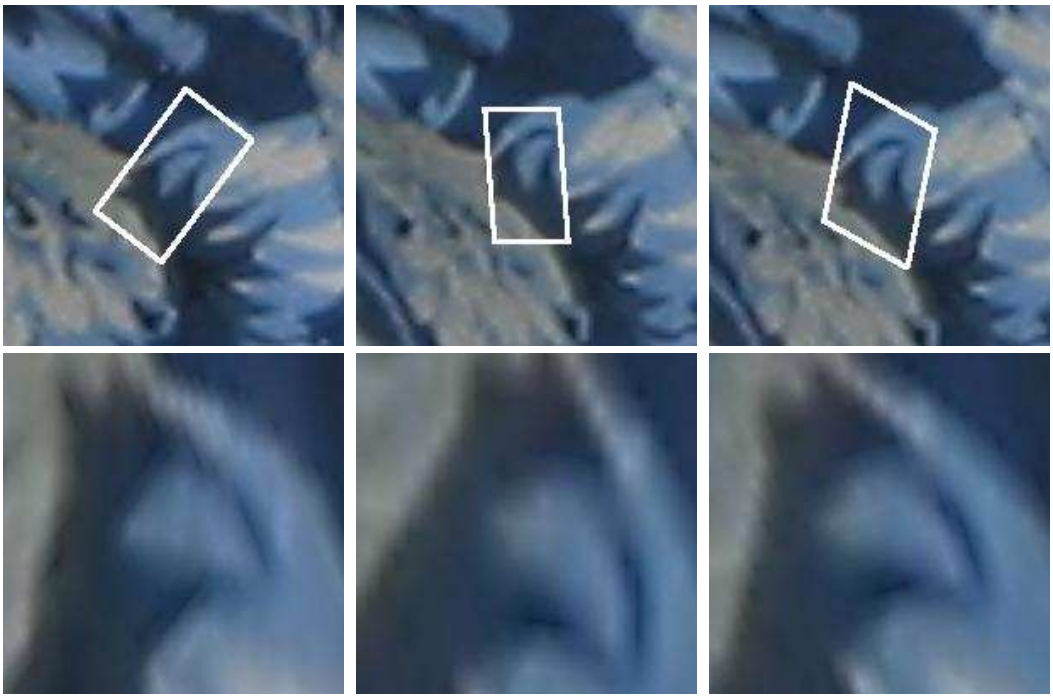


Figure 8. Adjusting the parameters of matched affine regions. Image patches are shown in the top part of the figure, and the corresponding rectified patches are shown in the bottom one. From left to right: The (constant) reference patch, and the variable patch before and after refinement. As expected, the rectified image patches are much closer to each other after refinement.

Once potential matches have been refined, we compare the paired patches by normalized correlation, and those that fall below a threshold of 0.9 are rejected. A simple neighborhood constraint is then used to further prune inconsistent ones: For a *primary* correspondence between image regions R_m and R_t to be retained, a sufficient fraction of the 10 nearest neighbors of R_m should also match neighbors of R_t. Call the number of these *secondary* matches the *score* of the primary correspondence they support. Since every affine region has roughly K potential matches, the score is bounded by $10K$. We retain correspondences whose score is at least two standard deviations above average. In a typical case (matching the first two bear images), the mean score is 1.2, with a standard deviation of 3.1. The threshold for retaining matches is thus 7.4, and 1,150 of the initial 16,800 correspondences are retained in this case.

3.1.2. *Robust Estimation*

The sampling and consensus parts of this procedure follow the steps described in Section 2.3. During sampling, factorization is used to solve Eq. (1) for the two projection matrices and the 3D configurations of the two sample patches. During consensus, the projection matrices are held constant, and the configuration of every patch added to the consensus set is estimated from Eq. (1) using linear least squares.

Similar approaches have of course been used before in the context of wide-baseline stereo, although the geometric constraints exploited in that case are usually related to the distance between matching points and the corresponding epipolar lines (Pritchett and Zisserman, 1998; Schaffalitzky and Zisserman, 2002; Baumberg, 2000; Tell and Carlsson, 2000; Matas et al., 2002; Tuytelaars and Van Gool, 2004). The reprojection error is a more natural metric in our context where two matching patches determine both the projection matrices and the 3D patch configurations, and it yields excellent results in practice. In our experiments, we have used both plain RANSAC and a variant where the samples are chosen in a deterministic, greedy fashion. Concretely, the greedy variant uses each potential match as a seed for a group, iteratively adding the match minimizing the mean reprojection error until this error exceeds 0.1 pixels,

or the group's size exceeds 20. In practice, both methods give almost identical results, RANSAC being slightly more efficient, and its greedy variant being slightly more reliable. The parameters used in our experiments are given in Figure 9, along with the computational costs for the two variants.

Method	Cost	K	M	N
RANSAC	$O(M\|P\|)$	[5,10]	1199	2
Greedy	$O(N\|P\|^2)$	[5,10]	$\|P\|$	20

Figure 9. Parameters for the two robust estimation strategies used to match pairs of images in our experiments, along with their combinatorial cost. Here $|P|$ denotes the size of the set P of match hypotheses, K is the number of best matches kept per model patch, M is the number of samples drawn, and N is the size of one seed. The value of M for RANSAC is based on an inlier rate of $w = 5\%$, M being chosen in this case as $E(M) + 2S(M)$, where $E(M) = w^{-N}$ is the expected value of the number of draws required to get one good sample and $S(M) = \sqrt{1 - w^N}/w^N$ is its standard deviation. See (Forsyth and Ponce, 2002, p. 347) for details.

We use a second neighborhood constraint to remove outliers at the end of this stage. It involves finding the five closest neighbors of a point in one image and the five closest neighbors of its putative match in the other image. If the match is consistent, the neighbors should also be matched with each other (barring occlusion). We test for this by comparing the barycentric coordinates[3] of the centers of matched regions relative to all $\binom{5}{3} = 10$ triples of their neighbors (Figure 10). The test is done symmetrically for the two images, and it examines 20 triples of neighbors. Two vectors of barycentric coordinates $\boldsymbol{x}$ and $\boldsymbol{y}$ are judged consistent if their relative distance $|\boldsymbol{x} - \boldsymbol{y}|/\max(|\boldsymbol{x}|, |\boldsymbol{y}|)$ is less than 0.5, and matches consistent with fewer than 8 of the 20 possible triples are rejected.

3.1.3. *Geometry-Based Addition of Matches*

The set of consistent matches found by the estimation step typically provide a good estimate of the epipolar geometry of the image pair. For each patch in the first image, we search for all patches in the second image whose "epipolar distance" is less than 2.5 pixels, and add up to K new matches. Specifically, we define the epipolar distance

[3] In a plane, the barycentric coordinates $(\alpha_1, \alpha_2, \alpha_3)$ of a point P in the basis formed by three other points A_1, A_2, and A_3 are uniquely defined by $\overrightarrow{OP} = \alpha_1\overrightarrow{OA}_1 + \alpha_2\overrightarrow{OA}_2 + \alpha_3\overrightarrow{OA}_3$, where O is an arbitrary point in the plane, and $\alpha_1 + \alpha_2 + \alpha_3 = 1$. These coordinates are independent of the choice of O, and invariant under affine transformations.

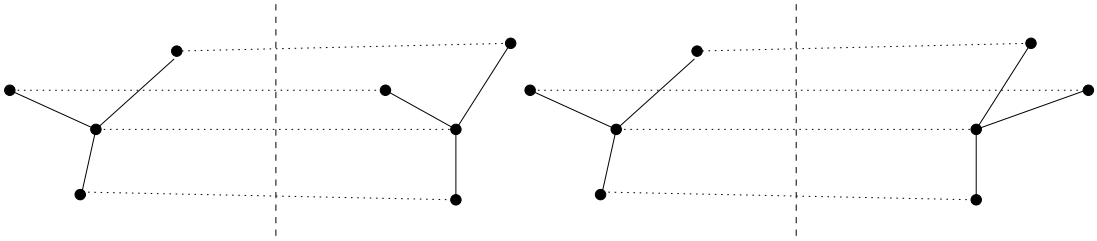


Figure 10. The barycentric neighborhood constraint. Left: Consistent matches. Right: Inconsistent ones.

as $d(\boldsymbol{c}_1, \mathcal{F}\boldsymbol{c}_2) + d(\boldsymbol{c}_2, \mathcal{F}^T\boldsymbol{c}_1)$, where $d(\boldsymbol{p}, \boldsymbol{l})$ gives the perpendicular distance between a point $\boldsymbol{p}$ and a line $\boldsymbol{l}$ in pixels, $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ are the patch centers in the two images, and $\mathcal{F}$ is the fundamental matrix.

3.2. Merging Partial Models into Composite Ones

The result of the image matching process is a collection of matches between neighboring training images (Figure 11). There are several combinatorial and geometric problems to solve in order to convert this information into a 3D model. The overall process is divided into four steps: (1) *chaining*: link matches across multiple images; (2) *stitching*: solve for the affine structure and motion while coping with missing data; (3) *bundle adjustment*: refine the model using non-linear least squares; and (4) *Euclidean upgrade*: use constraints associated with (partially) known intrinsic parameters of the camera to turn the affine reconstruction into a Euclidean one. The following sections describe each of these steps in detail.

3.2.1. *Chaining*

The matching process described in the previous section outputs affine regions matched across pairs of views. These matches can be represented in a single *match graph* structure, where each vertex corresponds to an affine region, labeled by the image where it was found, and arcs link matched pairs of regions. Intuitively, the set of views of the same surface patch forms a connected component of the match graph, which can in turn be used to form a sparse *patch-view* matrix whose columns represent surface patches, and rows represent the images in which they appear (Figure 12).

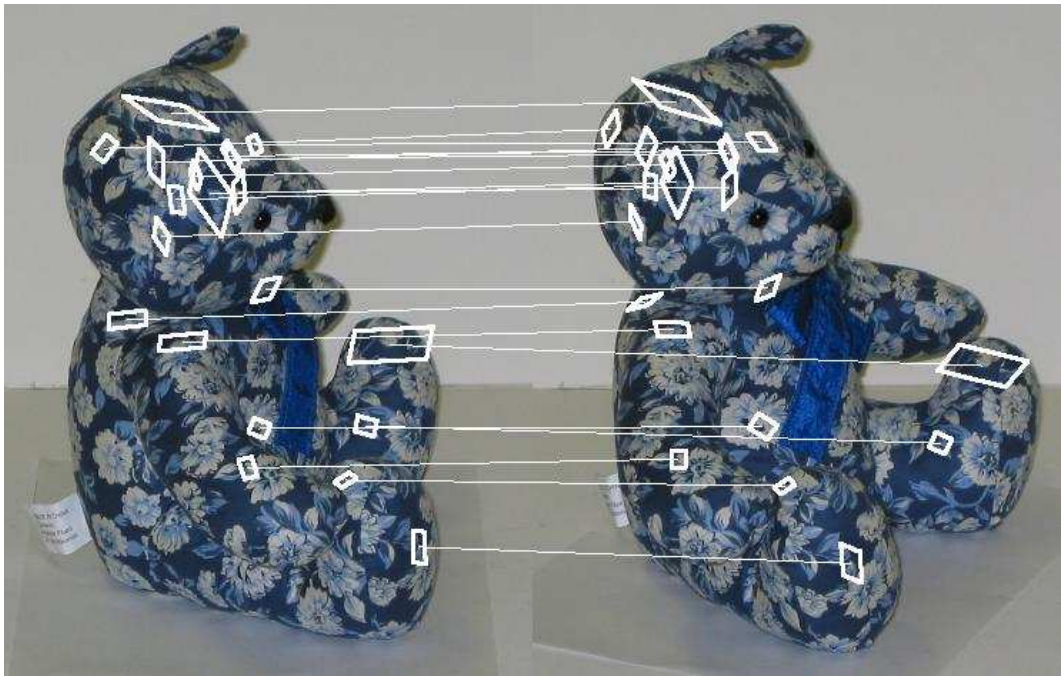


Figure 11. Matches between two images of the bear. For clarity, only 20 are shown.

Figure 12. A (subsampled) patch-view matrix for the teddy bear. The full patch-view matrix has 4,212 columns. Each black square indicates the presence of a given patch in a given image.

In practice, the construction of the patch-view matrix is complicated by the fact that different paths may link a vertex of the match graph to more than one vertex associated with a single view. We have chosen a simple heuristic to solve this problem: First, we associate with each connected component of the graph a root vertex corresponding to the affine region with maximum scale. Second, we refine the parameters of the region associated with every vertex in the connected component to maximize its correlation with the root, in much the same way as during image-to-image matching. This is necessary because some drift may be introduced in the parameters when chaining multiple views (Figure 13). Third, we enumerate all the vertices associated with each

image in the dataset, retain the representative vertex closest in feature space to the root vertex, and discard all others. This ensures that every image is represented by at most one vertex in each connected component, and affords a straightforward method for constructing the patch-view matrix.

Figure 13. Refining patch parameters across multiple views: Rectified patches associated with a match in four views before (top) and after (bottom) applying the refinement process. The patch in the rightmost column is the "root", and is used as a reference for the other three patches. The errors shown in the top row are exaggerated for the sake of illustration: The regions shown there are the unprocessed output of the affine region detector. In actual experiments, the refined parameters found during image matching are propagated along the edges of the match graph to provide better initial conditions.

3.2.2. *Stitching*

The patch-view matrix is comparable to the data matrix used in factorization approaches to affine structure from motion (Tomasi and Kanade, 1992). If all patches appeared in all views, we could indeed factorize the matrix directly to recover the patches' 3D configurations as well as the camera positions. In general, however, the matrix is sparse, and we must find dense blocks (submatrices) to factorize and stitch. The problem of finding maximal dense blocks of views and patches within the matrix reduces to the NP-complete problem of finding maximal cliques in a graph. In our implementation, we use a simple heuristic strategy which, while not guaranteed to be optimal or complete, generally produces an adequate solution: Briefly, we find a dense block for each patch—that is, for each column in the patch-view matrix—by searching for all other patches that are visible in at least the same views. In practice, this strategy

provides both a good coverage of the data by dense blocks, and an adequate overlap between blocks. Typically, patches appear in at least three or four views, depending on the separation between successive views in the sequence, and there are in general two orders of magnitude more patches than views.

The factorization technique described in Section 2.2.2 can of course be applied to each dense block to estimate the corresponding projection matrices and patch configurations in some local affine coordinate system (Figure 14). The next step is to combine the individual reconstructions into a coherent global model, or equivalently register them in a single coordinate system. With a proper set of constraints on the affine registration parameters, this can easily be expressed as an eigenvalue problem. In our experiments, however, we have found this linear approach to be numerically ill behaved (this is related to the inherent affine *gauge ambiguity* of our problem, see (Triggs et al., 1999) for a discussion of this issue). Thus, in practice, we pick an arbitrary block as *root*, and iteratively register all others with this one using linear least squares, before using a non-linear method to refine the global registration parameters.

We use the *stitch graph* to assist in this process. Its vertices are the blocks, and an edge between two vertices indicates that the corresponding blocks overlap. We choose the largest block as root node and use its coordinate system as the global frame. We then find the best path from the root to every other node using a measure that maximizes the number of points shared by adjacent blocks, the rationale being that large overlaps will give reliable estimates of the corresponding (local) registration parameters. Specifically, we assign to each edge a *capacity* (number of points common to the blocks associated with the incident vertices), and use a form of Dijkstra's algorithm to find for each vertex the path maximizing the capacity reaching the root.

The local registration parameters are concatenated along these paths, and they provide an estimate of the root-to-target affine transformation. Non-linear least-squares are finally used to minimize the mean-squared Euclidean distance between the centers of every pair of overlapping patches. After registering the blocks as described above, we combine all the camera and patch matrices into a single model. Since several

Figure 14. Sample partial models of the bear estimated from dense blocks. The blocks in this illustration were found by taking adjacent modeling views and selecting all patches they have in common. The partial models are all presented in a common coordinate frame, rather than in their local frames determined by factorization.

blocks may provide a value for a given camera or patch, we give preference to those closer to the root.

3.2.3. *Bundle Adjustment*

Once all blocks are registered, the initial estimates of the variables $\mathcal{M}_i$ and $\mathcal{N}_j$ are refined by minimizing

$$E = \sum_{j=1}^{n} \sum_{i \in I_j} |\mathcal{S}_{ij} - \mathcal{M}_i \mathcal{N}_j|^2, \tag{2}$$

where I_j denotes the set of images where patch number j is visible. Given the reasonable guesses available from the initial registration, this non-linear least-squares process only takes (in general) a few iterations to converge.

We have implemented two non-linear methods for minimizing the error E in Eq. (2). One is a sparse version of the Levenberg-Marquardt (LM) algorithm. The other uses a bilinear alternation strategy, that works by first holding the patches constant while solving for the cameras, then holding the cameras constant while solving for the patches, and iterating until convergence (see Mahamud et al. (2001) for a related approach to projective structure from motion). Note that the alternation strategy has first-order convergence properties, while LM has second-order convergence (Triggs et al., 1999). In general, LM requires fewer iterations than bilinear alternation, but its cost per iteration is much higher. For the size and density of the matrices typical of our modeling problems, we prefer the bilinear method, since in practice it finishes much sooner and produces essentially the same results as sparse LM.

The completed 3D model (Figure 15) consists of the matrices $\mathcal{M}_i$ and a description of each 3D surface patch j: the matrix $\mathcal{N}_j$ and the corresponding rectified texture patch. This patch can be constructed in a number of ways. One possibility is to combine the texture information from each measured image patch into a single high-quality copy using super-resolution techniques (Cheeseman et al., 1994; Capel and Zisserman, 2001; Baker and Kanade, 2002), provided the patches satisfy our assumption of planarity and that they are well registered. Currently, we simply choose the

image patch with the largest characteristic scale and copy its texture into the model. This is sufficient for the purpose of matching the model to novel images.

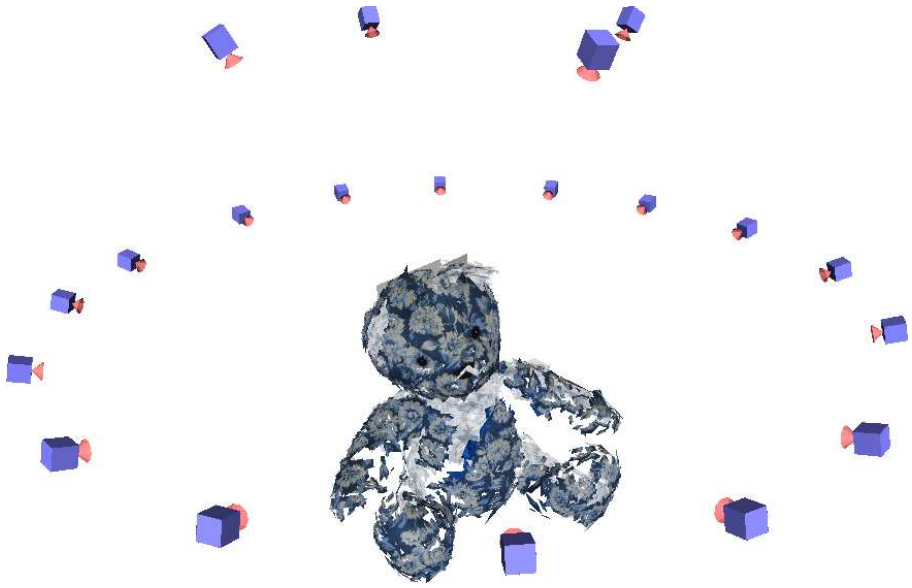


Figure 15. The bear model, along with the recovered affine camera configurations. These cameras are shown at an arbitrary constant distance from the origin.

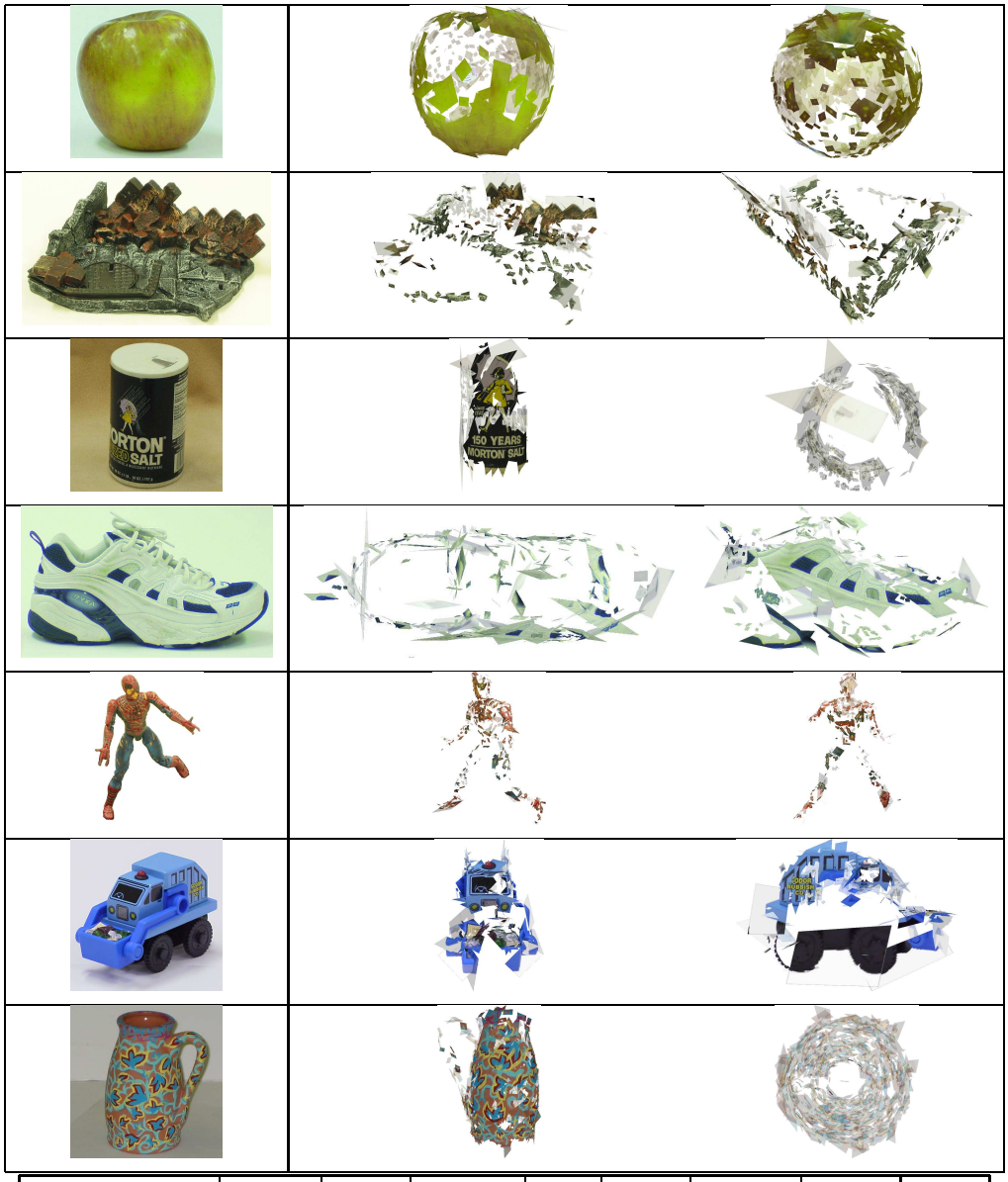
3.2.4. *Euclidean Upgrade*

It is not possible to go from affine to Euclidean structure and motion from two views only (Koenderink and van Doorn, 1991). When three or more views are available, on the other hand, it is a simple matter to compute the corresponding Euclidean weak-perspective projection matrices (assuming zero skew and known aspect-ratios) and recover the Euclidean structure (Tomasi and Kanade, 1992; Ponce, 2000): Briefly, we find the 3×3 matrix $\mathcal{Q}$ such that $\mathcal{A}_i\mathcal{Q}$ is part of a (scaled) rotation matrix for $i = 1, \ldots, m$. This provides linear constraints on $\mathcal{Q}\mathcal{Q}^T$, and allows the estimation of this symmetric matrix via linear least-squares. The matrix $\mathcal{Q}$ can then be computed via Cholesky decomposition for example (Poelman and Kanade, 1997; Weinshall and Tomasi, 1995).

The current implementation of our modeling approach is quite reliable, but rather slow: The teddy bear shown in Figure 15 is our largest model, with 4014 model patches computed from 20 images (24 image pairs). Image matching takes about 75 minutes per pair using pure RANSAC, for a total of 29.9 hours.[4] Image matching using the greedy algorithm takes 88 minutes per pair for a total of 35.2 hours. The final model is assembled from the partial ones in 1.5 hours. The greatest single expense in our modeling procedure is patch refinement. By selecting less stringent convergence criteria for this process and using a fixed 16×16 resolution for the image regions used to drive the LM procedure, it is possible to reduce the matching time to 6.6 minutes per image pair and assemble the model in 42 minutes, at the cost of getting 4% fewer 3D patches. Since modeling speed is not a priority in the context of this presentation, we have used the original refinement parameters in the rest of our experiments.

We have applied the modeling approach presented in this section to seven other objects, namely, an apple, the rubble-covered stand for a Spiderman action figure (called simply "rubble" from now on), a salt can, a shoe, Spidey himself, a toy truck, and a vase (Figure 16). For each object, the figure shows one sample from the set of input pictures. Each object model has been constructed using 16 to 20 input images, except for the apple which is modeled from 29 images to attain complete surface coverage. Beside each sample input image, the figure shows two renderings of the recovered Euclidean model. The models are rather sparse, but one should keep in mind that they are intended for object recognition, not for image-based rendering applications.

[4] All computing times in this presentation are given for C++ programs executed on a 3Ghz Pentium 4 running Linux.



	Apple	Bear	Rubble	Salt	Shoe	Spidey	Truck	Vase
Input images	29	20	16	16	16	16	16	20
Model patches	759	4014	737	866	488	526	518	1085

Figure 16. Object gallery. Left column: One of several input pictures for each object. Right column: Renderings of each model, not necessarily in same pose as input picture. Top to bottom: An apple, rubble (Spiderman base), a salt can, a shoe, Spidey, a toy truck, and a vase.

4. 3D Object Recognition

We now assume that the modeling approach presented in Section 3 has been used to create a library of 3D object models, and address the problem of identifying instances of these models in a test image. In many respects, this process is analogous to the method described in Section 3.1 for pairwise image matching. As before, Algorithm 1 outlines the overall process. Further details are given in the rest of this section.

4.1. Appearance-Based Selection of Potential Matches

Since matching is much more challenging in the recognition context where images may be heavily cluttered than in modeling tasks where there is essentially no clutter, we exploit both the SIFT descriptors and color histograms to select initial matches. More specifically, we use (1) a measure of the contrast (average squared gradient norm) in the patch, (2) a 10×10 color histogram drawn from the UV portion of YUV space, and (3) SIFT. To match feature vectors, we rely on color to filter out unpromising matches before comparing the remaining ones with SIFT. The level of contrast determines whether to use a tight or relaxed threshold on color.

We compare color histograms with the χ^2 metric, defined as

$$\sum_i \frac{(a_i - b_i)^2}{a_i + b_i},$$

where a_i and b_i are bins corresponding to each other in the respective histograms, and i iterates over the bins. The resulting value is in the $[0, 2]$ range, with 0 being a perfect match and 2 a complete mismatch.

Figure 17 illustrates the usefulness of multiple local image descriptors in matching tasks, particularly when the patches have low contrast. This example is taken from a test image for the apple. The model patch is in the center, the correct match is on the left, and an incorrect match is on the right. By human perception, all three patches appear almost identical, except that the incorrect patch has a different color. By SIFT distance, the incorrect match is actually closer than the correct one. The use of a color descriptor enables us to select the correct one.

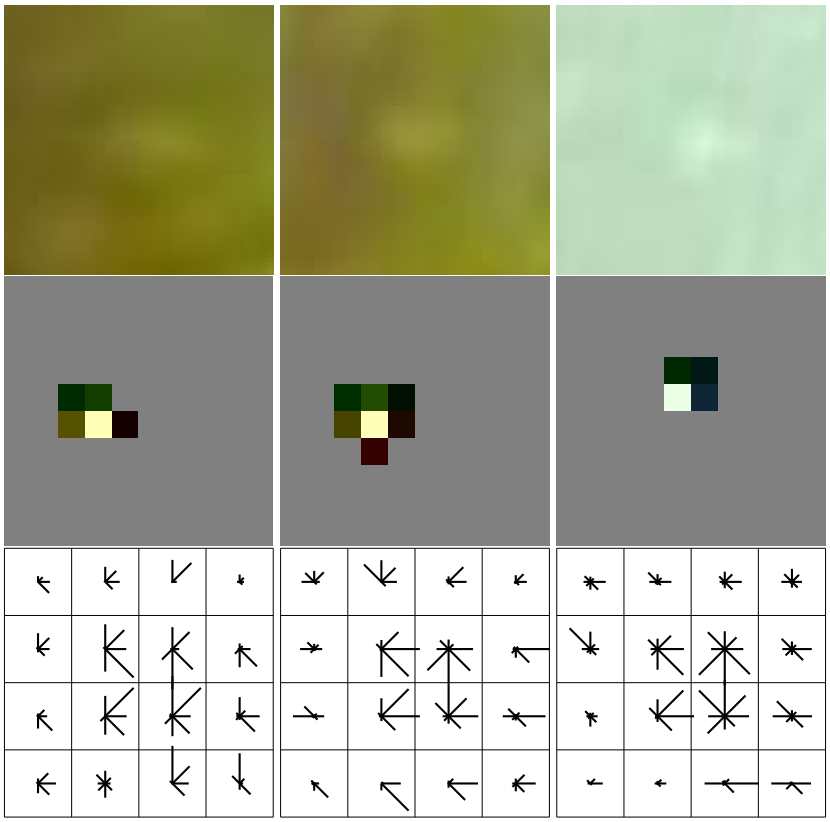


Figure 17. Comparing SIFT and color descriptors on low-contrast patches. The center column is the model patch. The left column is the correct match in the image. The right column is the match in the image ranked first by SIFT (but that is in fact an incorrect match). The top row shows the patch, the middle row shows the color histogram, and the bottom row shows the SIFT descriptor. The incorrect match has a Euclidean distance of 0.52 between SIFT descriptors and a χ^2 distance of 1.99 between the corresponding color histograms; and the correct match has a SIFT distance of 0.67 and a color distance of 0.03. The two patches on the left are red-green colored, while the patch on the right is aqua.

We use as before non-linear least squares to refine the parameters of the matched image regions to maximize their correlation with the corresponding model patches. Since this process is computationally expensive, we first apply a neighborhood constraint similar to that used in image matching to discard obviously inconsistent matches, as described next.

4.1.1. *Euclidean Neighborhood Constraints*

We saw earlier that affine models constructed from multiple views can be upgraded into Euclidean ones. In turn, a Euclidean model can be used to impose neighborhood constraints on individual matches: It is well known that three point matches—or in

our case, a single match between the corners and center of a model patch and those of an affine image region—are sufficient to determine the pose of a 3D object for calibrated cameras (Huttenlocher and Ullman, 1987). Thus, we recover the object pose associated with each potential match, and use it to reproject all other model patches into the image. Any patch whose reprojection falls close enough to a compatible affine region casts a vote for the match. Match candidates with above-average support are retained, and passed on to the refinement step.

In our implementation, the weight w of each vote depends on three factors, namely the characteristic scale σ_0 of the *primary* image region associated with the match candidate, the distance d between the projection of the voting patch and the corresponding *secondary* image region, and the distance d_0 between the primary and secondary regions. In practice, we set $w = G_\sigma(d)$, where G_σ is a Gaussian distribution with standard deviation $\sigma = 10 + d_0/4\sigma_0$ (Figure 18). With this choice, small values of d correspond to large votes, and the contribution of each secondary patch is modulated so the Gaussian sharply peaks for large primary regions likely to yield accurate pose estimates, and for secondary regions more likely to be accurately localized because they are close to the primary ones.

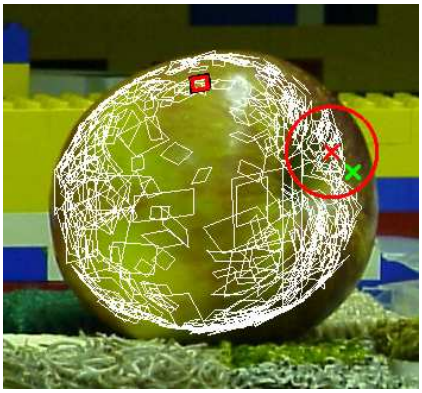


Figure 18. An illustration of the proposed voting scheme: The primary match that determines the pose appears as a heavy parallelogram, and all the forward facing patches projected from the model appear as light parallelograms. The projected center of the supporting match appears as an "$\times$" surrounded by a circle. The actual image position of the supporting match appears as another "$\times$". The radius of the circle is equal to the standard deviation of the Gaussian distribution deciding the weight of the corresponding vote.

As noted in Section 2, various methods for finding matching features consistent with a given set of geometric constraints have been proposed in the past, including interpretation tree—or alignment—techniques (Ayache and Faugeras, 1986; Faugeras and Hebert, 1986; Grimson and Lozano-Pérez, 1987; Huttenlocher and Ullman, 1987; Lowe, 1987), geometric hashing (Lamdan and Wolfson, 1988; Lamdan and Wolfson, 1991), and robust statistical methods such as RANSAC (Fischler and Bolles, 1981) and its variants (Torr and Zisserman, 2000). Both alignment and RANSAC can easily be implemented in the context of Algorithm 1. We have experimented with several alternatives: The first one is a recursive implementation of alignment where an interpretation tree is visited in a depth-first manner (*null* matches between model patches and "empty" image regions being used to handle occlusion and faulty detection) until a maximum depth N is reached ($N = 20$ in our experiments), or the mean reprojection error exceeds 1 pixel in all branches up to that depth (see Ayache and Faugeras, 1986; Faugeras and Hebert, 1986 for more details on this approach). We have also implemented plain RANSAC and two variants: a "greedy" version where, as before, M groups of matches of size lesser than or equal to N are chosen in a deterministic, greedy manner to minimize the mean projection error, and used instead of random samples; and an "exhaustive" version where all pairs of candidate matches are examined. The computational costs of the RANSAC variants are easy to estimate, and they are given in Figure 19. The cost of alignment is more difficult to assess, but can be shown to be a low-order polynomial in the size n of the model when there is little or no clutter, and exponential in n in the presence of clutter when no limit on the depth of the tree search is imposed (Grimson, 1990). The worst-case computational complexity of our bounded tree search is $O(n^N)$, but determining its expected cost is beyond the scope of this paper. As will be shown in Section 4.5, the "greedy" version of RANSAC has performed best in our experiments.

Method	Cost	K	M	N
RANSAC	$O(M\|P\|)$	L/n	$[1998, 12498]$	2
Alignment	see Sec. 4.2	L/n	n	20
Exhaustive	$O(\|P\|^3)$	L/n	$\|P\|^2$	2
Greedy	$O(N\|P\|^2)$	L/n	$\|P\|$	20

Figure 19. Parameters for the different geometric estimation methods for Algorithm 1 used in our recognition experiments, along with their combinatorial cost. Here, L denotes a preset number of potential matches to be examined ($L = 12,000$ in our experiments), and n is the number of patches per object model.

4.3. GEOMETRY-BASED ADDITION OF MATCHES

The matches found by the estimation step provide a projection matrix that places the model into the image. All forward facing patches in the model could potentially be present in the image. Therefore, we project each such model patch and select the K closest image patches as new match hypotheses.

4.4. OBJECT DETECTION

Once an object model has been matched to an image, some criterion is needed to decide whether it is present or not. After experimenting with a few reasonable choices, we have settled on the following criterion:

$$(\text{number of matches} \geq m \text{ OR matched area/total area} \geq a) \text{ AND distortion} \leq d,$$

where nominal values for the parameters are $m = 10$, $a = 0.1$, and $d = 0.15$. Here, the measure of distortion is

$$\frac{\boldsymbol{a}_1^T \boldsymbol{a}_2}{|\boldsymbol{a}_1||\boldsymbol{a}_2|} + \left(1 - \frac{\min(|\boldsymbol{a}_1|, |\boldsymbol{a}_2|)}{\max(|\boldsymbol{a}_1|, |\boldsymbol{a}_2|)}\right),$$

where $\boldsymbol{a}_i^T$ is the ith row of the leftmost 2×3 portion $\mathcal{A}$ of the projection matrix, and it reflects how close to the top part of a scaled rotation this matrix is. The matched surface area of the model is measured in terms of the patches whose normalized correlation is above the usual thresholds, and it is compared to the total surface area actually visible from the predicted viewpoint. The influence of the three parameters on recognition performance is studied in the next section.

Our recognition experiments match all eight of our object models against a set of 51 images (the photograph from Figure 1 and the 50 pictures shown in Figure 20). Each image contains instances of up to five object models, even though most of them only contain one or two. Figure 21 gives quantitative recognition results for the different monochrome variants of our algorithm, where color information is not used. The parameters for these tests are fixed to their nominal values of $m = 10$, $a = 0.1$, and $d = 0.15$. With these settings, none of the methods tested gives false positives, and the "greedy" version of RANSAC with $N = 20$ gives the best performance, with a recognition rate (averaged over the eight object models) of 88%. The time costs given in the table are per image-object combination, in minutes.

Since it has consistently performed best in our experiments, we will from now on focus on the greedy variant of RANSAC with $N = 20$. It is interesting to compare different image descriptors and to test whether the use of color information may boost recognition performance. Figure 22 shows the results of a quantitative experiment: It can be seen that the combination of color and SIFT gives the best performance, with a mean recognition rate of 94%. (This rate is for the nominal settings of the detection parameters. The effect of these parameters is discussed below.) Using color together with plain patch correlation results in performance similar to that of SIFT descriptors without color information.

As is always the case in object recognition, many implementation parameters can be varied in our program: For example, Figure 23 shows the trade-off between computing cost and recognition accuracy that can be achieved by changing the patch size used to refine the alignment between matched affine regions. As shown by this figure, selecting a fixed 16×16 resolution instead of the original resolution of the test patch used in the previous experiments halves the computing time with essentially no effect on recognition accuracy. Lowering the resolution too much, on the other hand, clearly affects recognition performance.

Figure 20. The dataset (51 images) used in our recognition experiments: 50 of the images are shown here. The last one is shown in Figure 1.

Method	Apple 11	Bear 11	Rubble 9	Salt 10	Shoe 9	Spidey 4	Truck 12	Vase 12	Mean	Time
RANSAC	3	11	8	9	2	3	9	11	71%	4.3
Alignment	5	10	9	10	4	4	12	12	85%	7.5
Exhaustive	5	11	9	10	4	4	12	12	86%	7.7
Greedy ($N = 2$)	6	11	9	10	3	4	12	12	86%	5.9
Greedy ($N = 20$)	5	11	9	10	5	4	12	12	88%	6.7

Figure 21. Comparison of recognition rates for different monochrome variants of our method. See text for details. The row of numbers immediately under the object names gives the true number of instances present in the test images.

Method	Apple 11	Bear 11	Rubble 9	Salt 10	Shoe 9	Spidey 4	Truck 12	Vase 12	Mean	Time
Correlation	6	11	8	10	4	4	10	8	80%	5.6
SIFT	5	11	9	10	5	4	12	12	88%	6.7
Correlation + Color	8	11	9	10	6	4	10	11	89%	3.9
SIFT + Color	8	11	9	10	7	4	12	12	94%	3.7

Figure 22. Comparison of recognition rates for different descriptors using the greedy RANSAC variant with $N = 20$.

The recognition rates reported so far are for fixed, nominal values of the detection parameters m, a, and d. A better understanding of our algorithm's performance can be gained by plotting the overall rates of true positives (instances where an object is correctly identified in an image) and true negatives (instances where an object is correctly determined to be absent) against a range of parameter values. Figure 24 shows the corresponding plots for the color version of our algorithm, where we vary one of the three parameters while holding the other two constant at their nominal values.

As shown by Figure 24, the recognition performance is quite stable over a reasonable range of detection parameters. The equal-error-rate parameter values correspond

Method	Apple 11	Bear 11	Rubble 9	Salt 10	Shoe 9	Spidey 4	Truck 12	Vase 12	Mean	Time
Original resolution	8	11	9	10	7	4	12	12	94%	3.7
16×16 resolution	8	11	9	10	7	4	12	12	94%	1.9
8×8 resolution	9	11	9	10	5	4	11	12	91%	1.6

Figure 23. Effect of region sampling during patch refinement on computation cost and recognition accuracy.

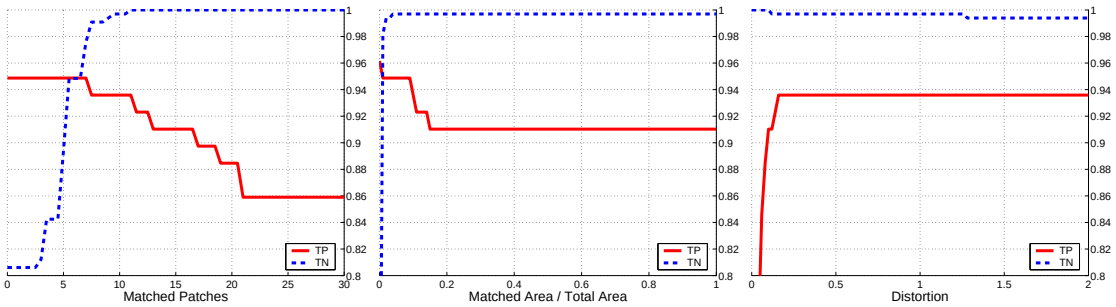


Figure 24. Dependency of the recognition rate on the detection parameters: The true positive (TP) and true negative (TN) rates are plotted by holding two of the detection parameters constant at their nominal values and varying, from left to right, the number of matched patches, the ratio of matched to visible area, and the distortion.

to the point (if any) where the true positive and true negative curves cross, which occurs in the 94–96% range in these graphs. The best recognition rate that we have been able to obtain by tuning the detection parameters is 95% with no false positives.

In order to obtain a quantitative comparison of our method with other state-of-the-art object recognition systems, we have provided our dataset[5] to several other research groups. The algorithms proposed by Ferrari, Tuytelaars & Van Gool (2004), Lowe (2004), Mahamud & Hebert (2003), and Moreels, Maire & Perona (2004) have been tested by their authors in this comparative study. As shown by Figure 25, all the algorithms perform well on our data set, achieving recognition rates of 90% and above for false detection rates below 10%. In this experiment, the color version of our algorithm and Lowe's (2004) program perform best for very low false detection rates, followed by the black-and-white version of our algorithm. The technique proposed by Ferrari et al. (2004) achieves an extremely high recognition rate at the cost of a somewhat higher false detection rate. Although all five algorithms use multiple views to form object models, only Lowe's algorithm and ours actually combine the information associated with multiple views in the recognition process.[6] The other methods consider all training pictures independently, which essentially reduces object recognition to image matching. The five algorithms use different geometric constraints to

[5] The data is publicly available at `http://www-cvr.ai.uiuc.edu/ponce_grp/data`.

[6] Lowe's algorithm does not construct an explicit 3D model, but it allows multiple training views sharing common patches to vote for the same object (Lowe, 2004).

reject inconsistent matches: We exploit the global 3D (affine and Euclidean) rigidity of our object models. Ferrari et al. (2004) use instead a set of *local* 2D affine rigidity constraints, which are somewhat weaker but allow the recognition of deformable objects such as magazines, and the remaining authors exploit *global* 2D (affine or Euclidean) rigidity constraints, best suited to situations where the training and test views are close to each other, or the relief of the scene is small compared to the distance separating it from the observer. To test the power of these constraints, we have included in our comparative study a baseline recognition method where the pairwise image matching part of our modeling algorithm is used as a simple recognition engine, an object being declared as recognized when a sufficient percentage of the patches founds in a training view are matched to the test image. The geometric constraints used in this case are quite weak, and amount to exploiting the epipolar geometry conventionally used in wide-baseline stereo. As shown by Figure 25, although this simple method gives reasonable results (over 50% true positive rate with no false positives), it gives the worse recognition rates of all methods tested.

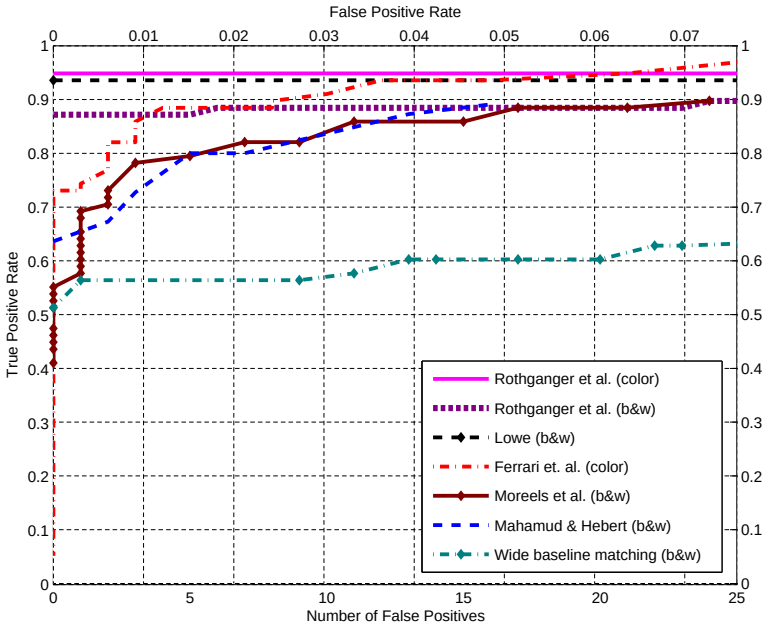


Figure 25. True positive rate plotted against number of false positives for several different recognition methods. For our curve, the three recognition parameters m, a, and d assume their best values for each level of false positives.

These results should not be interpreted as a conclusive ranking of the tested algorithms, since our test dataset is quite small, and it is probably biased in favor of our method. However, they provide some evidence (and this should not be particularly surprising) that combining multiple views improves recognition performance, and so does the inclusion of geometric constraints in the matching process. Of course, there is a price to pay for the integration of multiple images into a single model: First, this makes modeling more costly and complicated. Second, this requires the use of training views with sufficient overlap, as confirmed by our experiments with the data of Ferrari et al. (2004), where the input images have too few patches in common to allow us to construct any meaningful model.

Let us conclude with some qualitative experimental results, using as before the color/SIFT greedy variant of RANSAC with $N = 20$. Figure 26 shows sample results of some challenging—yet successful—recognition experiments, with a large degree of occlusion and clutter. Figure 27 shows the images where recognition fails. Very little of the apple is visible in two of the images where our program fails to recognize it, and highlights dominate its third picture. Maybe more surprisingly, the shoe occupies a large portion of the two images where it escapes detection. The reason is simply that we did not include overhead views of the shoe in the training set.[7] The shoe images shown in Figure 27 are separated by about 60° from the views used during modeling, with very few of the model patches appearing in the test pictures, which explains our program's failure and illustrates its limitations.

5. Discussion

We have proposed in this article to revisit invariants as a local object description that exploits the fact that smooth surfaces are always planar in the small. Combining this idea with the affine regions of Mikolajczyk and Schmid (2002) has allowed us to construct a normalized representation of local surface appearance that can be used to

[7] The shoe, like the apple, is now long gone, preventing us from adding any more training images.

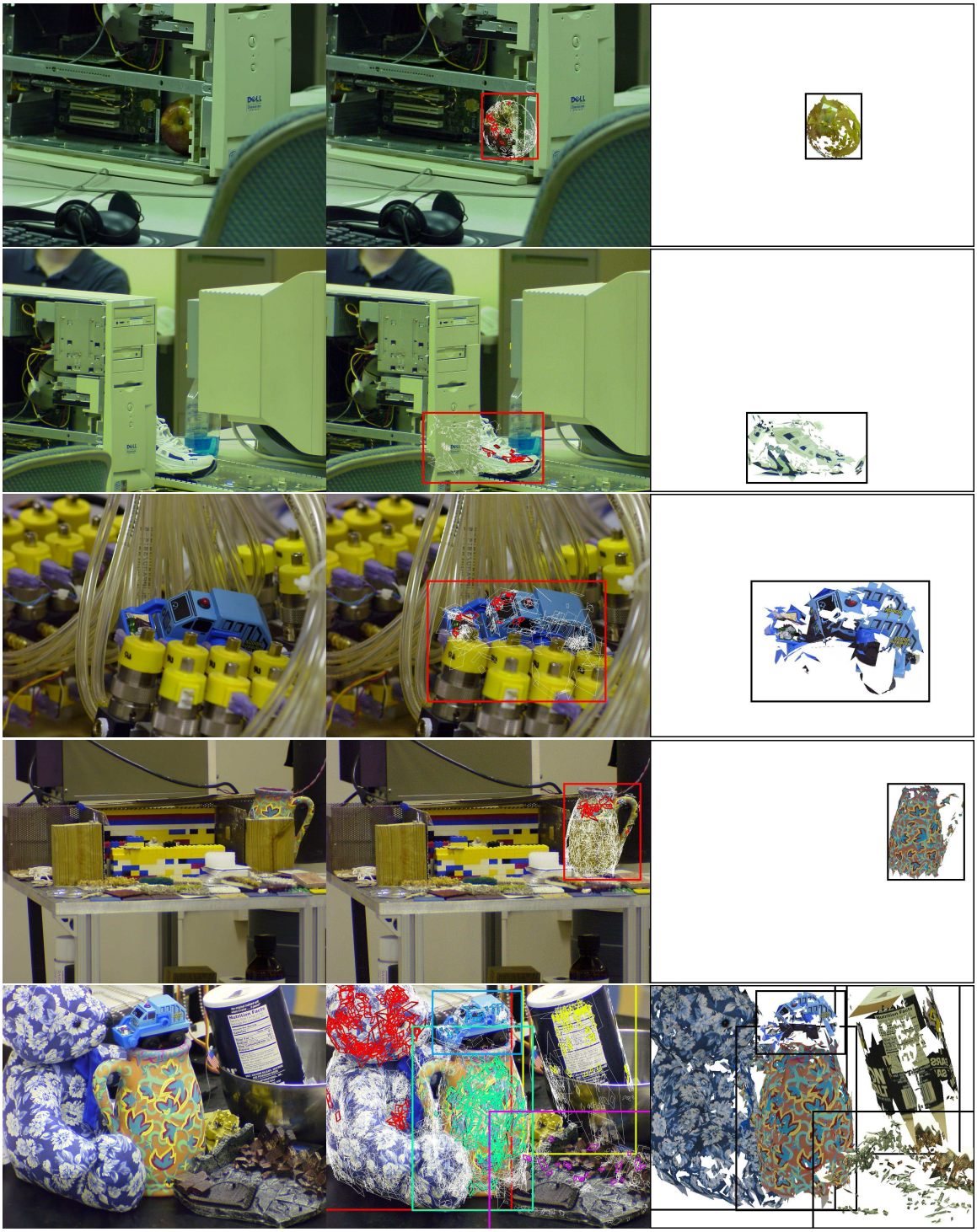


Figure 26. Some challenging but successful recognition results. As in Figure 1, the recognized models are rendered in the poses estimated by our program, and bounding boxes for the reprojections are shown as rectangles.

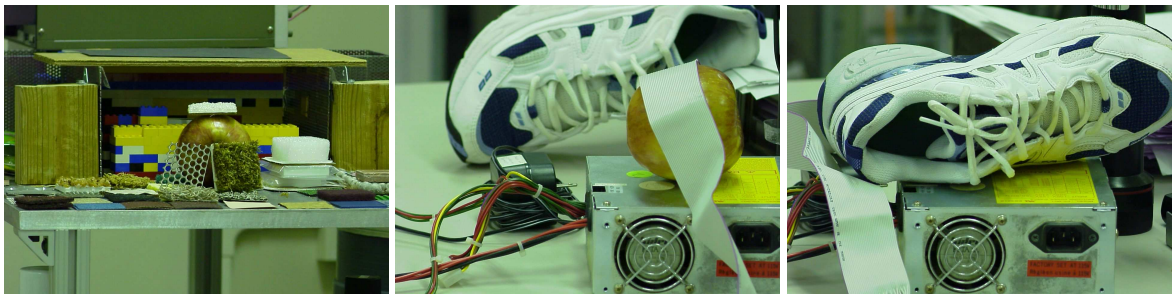


Figure 27. Closeups of the images where recognition fails.

select promising matches in 3D object modeling and recognition tasks. We have used multi-view geometric constraints to represent the larger 3D surface structure, retain groups of consistent matches, and reject incorrect ones. Our experiments demonstrate the promise of the proposed approach to 3D object recognition.

Our current implementation is limited to affine viewing conditions. As noted in Section 2.2, a match between $m \geq 2$ affine regions is equivalent to a match between m triples of points, thus the machinery developed in the structure from motion (Faugeras et al., 2001; Hartley and Zisserman, 2000; Tomasi and Kanade, 1992) and pose estimation (Huttenlocher and Ullman, 1987; Lowe, 1987) literature can in principle be used to extend our approach to the perspective case. This is particularly relevant in the context of scene interpretation (as opposed to individual object recognition), where the relief of each surface patch may be small compared to the overall depth of the scene, so that an affine projection model is appropriate for each patch, yet a global affine projection model is inappropriate (think of street scenes, for example, that exhibit significant perspective distortions). As a first step toward tackling this problem, we have recently introduced a local affine viewing model obtained by linearizing the perspective projection equations in the neighborhood of each patch, and used it to extend the approach proposed in this article to the problems of motion segmentation, scene modeling, and scene recognition in video clips (Rothganger et al., 2004).

Admittedly, our current implementation is slow, especially compared to the systems proposed by Lowe (2004), and Mahamud and Hebert (2003), that achieve framerate object detection in cluttered scenes. Speed was never our priority (despite some

efforts at optimizing our code), and we believe that our approach can (and should) be sped up by at least an order of magnitude using a more careful implementation. Two key changes would be to use a voting scheme rather than a full comparison of each object with each image, and to avoid patch refinement if possible.

An obvious limitation of our approach is its reliance on texture: Some objects (e.g., statues, cars, many kinds of fruit and vegetables) are essentially textureless, yet easily recognizable (for humans). Alternatively, many objects are heavily textured, but the corresponding patterns may be more distracting than characteristic (e.g., a cat's fur may look like a patchwork of different colors, it may sport stripes, or just be plain black or white, yet a person will still recognize the cat in the picture). Handling such objects will require new image descriptors that better convey shape (as opposed to appearance) information, yet capture an appropriate level of viewpoint invariance. Developing these descriptors and the corresponding recognition strategies is next on our agenda.

Acknowledgments. This research was partially supported by the National Science Foundation under grants IIS-0308087 and IIS-0312438, Toyota Motor Corporation, the UIUC-CNRS Research Collaboration Agreement, the European FET-open project VIBES, the UIUC Campus Research Board, and the Beckman Institute. We would like to thank V. Ferrari, M. Hebert, D. Lowe, S. Mahamud, M. Maire, P. Moreels, M. Munich, P. Perona, T. Tuytelaars, and L. Van Gool for kindly accepting to participate in the comparative study reported in Section 4.5. We would also like to thank A. Kushal for his help with our experiments.

Appendix A: Inverse Projection Matrices

Let us introduce more formally the inverse projection matrix associated with a plane under affine projection.

Consider a plane Π with coordinate vector $\mathbf{\Pi}$ in the world coordinate system. For any point in this plane we can write the affine projection in some image plane as

$\boldsymbol{p} = \mathcal{M}\boldsymbol{P}$ and $\boldsymbol{\Pi}^T\boldsymbol{P} = 0$. These two equations determine the homogeneous coordinate vector $\boldsymbol{P}$ up to scale. To completely determine it, we can impose that its fourth coordinate be 1, and the corresponding equations become

$$\mathcal{M}_{\boldsymbol{\Pi}}\boldsymbol{P} = \begin{bmatrix} \mathcal{M} \\ \boldsymbol{\Pi}^T \\ 0\ 0\ 0\ 1 \end{bmatrix} \boldsymbol{P} = \begin{bmatrix} \boldsymbol{p} \\ 0 \\ 1 \end{bmatrix}.$$

Not surprisingly, $\mathcal{M}_{\boldsymbol{\Pi}}$ is an affine transformation matrix. So is its inverse, and if

$$\mathcal{M}_{\boldsymbol{\Pi}}^{-1} = \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \boldsymbol{c}_3 & \boldsymbol{c}_4 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

we can write

$$\boldsymbol{P} = \mathcal{M}_{\boldsymbol{\Pi}}^{-1} \begin{bmatrix} \boldsymbol{p} \\ 0 \\ 1 \end{bmatrix} = \mathcal{M}_{\boldsymbol{\Pi}}^{\dagger} \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix}, \text{ where } \mathcal{M}_{\boldsymbol{\Pi}}^{\dagger} \stackrel{\text{def}}{=} \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \boldsymbol{c}_4 \\ 0 & 0 & 1 \end{bmatrix}.$$

The 4×3 matrix $\mathcal{M}_{\boldsymbol{\Pi}}^{\dagger}$ is the *inverse projection matrix* (Faugeras et al., 2001) associated with the plane Π. Note that, for any point $\boldsymbol{p}$ in the image plane, the point

$$\boldsymbol{P} = \mathcal{M}_{\boldsymbol{\Pi}}^{\dagger} \begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix}$$

lies in the plane Π, thus $\boldsymbol{\Pi}^T\boldsymbol{P} = 0$. Since this must be true for all points $\boldsymbol{p}$, we must have $\boldsymbol{\Pi}^T\mathcal{M}_{\boldsymbol{\Pi}}^{\dagger} = \mathbf{0}^T$.

The matrix $\mathcal{N}_j$ used in this paper is simply $\mathcal{M}_{\boldsymbol{\Pi}_j}^{(j)\dagger}$ where $\mathcal{M}^{(j)}$ is the matrix associated with the projection into the (fictitious) rectified image plane. Note that $\mathcal{M}^{(j)}$ maps the center C_j of patch number j onto the origin of the rectified image plane. It follows that the coordinate vector of this point is

$$\begin{bmatrix} \boldsymbol{C}_j \\ 1 \end{bmatrix} = \mathcal{N}_j \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix},$$

or, equivalently, that $\begin{bmatrix} \boldsymbol{C}_j \\ 1 \end{bmatrix}$ is the third column of the matrix $\mathcal{N}_j$. Similar reasoning shows that the "horizontal" and "vertical" axes of the patch are respectively the first and second columns of $\mathcal{N}_j$. Finally, we write the inverse projection matrix as

$$\mathcal{N}_j = \begin{bmatrix} \boldsymbol{H}_j & \boldsymbol{V}_j & \boldsymbol{C}_j \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathcal{B}_j \\ 0\ 0\ 1 \end{bmatrix},$$

where $\mathcal{B}_j$ is a 3×3 matrix.

Appendix B: Patch Refinement

We use the Levenberg-Marquardt (LM) non-linear least squares algorithm to do the alignment. Here we give the error function being minimized and show how to compute its Jacobian analytically. Let $P(\boldsymbol{x})$ be pixel values from the image containing the variable patch, and let $R(\boldsymbol{u})$ be pixel values from the normalized form of the fixed ("reference") patch, where $\boldsymbol{x}$ and $\boldsymbol{u}$ are homogeneous coordinates with scale fixed at 1. Let $\mathcal{S}$ be the inverse rectification matrix associated with the variable patch. The mapping function between the patches is

$$\boldsymbol{x} = \mathcal{S}\boldsymbol{u} = \begin{bmatrix} u_1 S_{11} + u_2 S_{12} + S_{13} \\ u_1 S_{21} + u_2 S_{22} + S_{23} \\ 1 \end{bmatrix} \tag{3}$$

We want to minimize the error

$$E = \sum_{\boldsymbol{u} \in R} |P(\mathcal{S}\boldsymbol{u}) - R(\boldsymbol{u})|^2,$$

with respect to $\mathcal{S}$. The error function for one pixel position $\boldsymbol{u}$ is then $e(\boldsymbol{u}) = P(\mathcal{S}\boldsymbol{u}) - R(\boldsymbol{u})$. The error function given to LM is the vector of $e(\boldsymbol{u})$ values produced by iterating $\boldsymbol{u}$ over all the discrete pixel positions in the reference patch. The parameters that LM modifies are the six elements $\mathcal{S}_{kl}$. We compute the elements of the Jacobian as

$$\frac{\partial e}{\partial \mathcal{S}_{kl}}(\boldsymbol{u}) = \frac{\partial P}{\partial x_1}\frac{\partial x_1}{\partial \mathcal{S}_{kl}} + \frac{\partial P}{\partial x_2}\frac{\partial x_2}{\partial \mathcal{S}_{kl}}.$$

Notice that the second term $R(\boldsymbol{u})$ in the function $e(\boldsymbol{u})$ drops out because it is constant w.r.t. $\mathcal{S}$. Also note that due to the form of the matrix multiplication in (3), only one of the two partial derivatives w.r.t. $\mathcal{S}_{kl}$ on the right is nonzero for any given subscript kl.

All that remains is to compute the partial derivatives $\partial P/\partial x_1$ and $\partial P/\partial x_2$ of P w.r.t. to the components of $\boldsymbol{x}$. A low cost way to approximate these is to take the pixel values p_{00}, p_{01}, p_{10} and p_{11} from the four discrete locations closest to $\boldsymbol{x}$ in P and

compute the slope by interpolation. For example, if $d = x_2 - \lfloor x_2 \rfloor$, we have

$$\frac{\partial P}{\partial x_1} = (1-d)(p_{01} - p_{00}) + d(p_{11} - p_{10}).$$

The expression for $\partial P / \partial x_2$ is similar.

LM will of course only find a local minimum of the error function rather than its global minimum. In practice, the initial guess from affine adaptation is generally close enough to the correct value for this method to give quite good results.

References

Ayache, N. and O. D. Faugeras: 1986, 'Hyper: a new approach for the recognition and positioning of two-dimensional objects'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **8**(1), 44–54.

Baker, S. and T. Kanade: 2002, 'Limits on Super-Resolution and How to Break Them'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(9), 1167–1183.

Baumberg, A.: 2000, 'Reliable Feature Matching Across Widely Separated Views'. In: *Conference on Computer Vision and Pattern Recognition*. pp. 774–781.

Belhumeur, P. N., J. P. Hespanha, and D. J. Kriegman: 1997, 'Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(7), 711–720.

Blostein, D. and N. Ahuja: 1989, 'A Multiscale Region Detector'. *Computer Vision, Graphics and Image Processing* **45**, 22–41.

Burns, J. B., R. S. Weiss, and E. M. Riseman: 1993, 'View Variation of Point-Set and Line-Segment Features'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(1), 51–68.

Capel, D. and A. Zisserman: 2001, 'Super-resolution from multiple views using learnt image models'. In: *Conference on Computer Vision and Pattern Recognition*.

Cheeseman, P., B. Kanefsky, R. Kraft, and J. Stutz: 1994, 'Super-Resolved Surface Reconstruction from Multiple Images'. Technical report, NASA Ames Research Center.

Crowley, J. L. and A. C. Parker: 1984, 'A representation of shape based on peaks and ridges in the difference of low-pass transform'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, 156–170.

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

Faugeras, O., Q. T. Luong, and T. Papadopoulo: 2001, *The Geometry of Multiple Images*. MIT Press.

Faugeras, O. D. and M. Hebert: 1986, 'The representation, recognition, and locating of 3-D objects'. *International Journal of Robotics Research* **5**(3), 27–52. 1986.

Fergus, R., P. Perona, and A. Zisserman: 2003, 'Object class recognition by unsupervised scale-invariant learning'. In: *Conference on Computer Vision and Pattern Recognition*, Vol. II. pp. 264–270.

Ferrari, V., T. Tuytelaars, and L. Van Gool: 2004, 'Simultaneous Object Recognition and Segmentation by Image Exploration'. In: *European Conference on Computer Vision*.

Fischler, M. A. and R. C. Bolles: 1981, 'Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography'. *Communications ACM* **24**(6), 381–395.

Forsyth, D. and J. Ponce: 2002, *Computer Vision: A Modern Approach*. Prentice-Hall.

Gårding, J. and T. Lindeberg: 1996, 'Direct computation of shape cues using scale-adapted spatial derivative operators'. *International Journal of Computer Vision* **17**(2), 163–191.

Grimson, W. E. L.: 1990, 'The combinatorics of object recognition in cluttered environments using constrained search'. *Artificial Intelligence Journal* **44**(1-2), 121–166.

Grimson, W. E. L. and T. Lozano-Pérez: 1987, 'Localizing Overlapping Parts by Searching the Interpretation Tree'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **9**(4), 469–482.

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

Hartley, R. and A. Zisserman: 2000, *Multiple view geometry in computer vision*. Cambridge University Press.

Huttenlocher, D. P. and S. Ullman: 1987, 'Object recognition using alignment'. In: *International Conference on Computer Vision*. pp. 102–111.

Kadir, T. and M. Brady: 2001, 'Scale, Saliency and Image Description'. *International Journal of Computer Vision* **45**(2), 83–105.

Koenderink, J. J. and A. J. van Doorn: 1991, 'Affine structure from motion'. *Journal of the Optical Society of America* **8**(2), 377–385.

Lamdan, Y. and H. J. Wolfson: 1988, 'Geometric Hashing: A General and Efficient Model-Based Reconitiion Scheme'. In: *International Conference on Computer Vision*. pp. 238–249.

Lamdan, Y. and H. J. Wolfson: 1991, 'On the Error Analysis of 'Geometric Hashing". In: *Conference on Computer Vision and Pattern Recognition*. Maui, Hawaii, pp. 22–27.

Lindeberg, T.: 1998, 'Feature Detection with Automatic Scale Selection'. *International Journal of Computer Vision* **30**(2), 77–116.

Liu, J., J. Mundy, D. Forsyth, A. Zisserman, and C. Rothwell: 1993, 'Efficient recognition of rotationally symmetric surfaces and straight homogeneous generalized cylinders'. In: *Conference on Computer Vision and Pattern Recognition*. New York City, NY, pp. 123–128.

Lowe, D.: 2004, 'Distinctive image features from scale-invariant keypoints'. *International Journal of Computer Vision*. In press.

Lowe, D. G.: 1987, 'The Viewpoint Consistency Constraint'. *International Journal of Computer Vision* **1**(1), 57–72.

Mahamud, S. and M. Hebert: 2003, 'The Optimal Distance Measure for Object Detection'. In: *Conference on Computer Vision and Pattern Recognition*.

Mahamud, S., M. Hebert, Y. Omori, and J. Ponce: 2001, 'Provably-Convergent Iterative Methods for Projective Structure from Motion'. In: *Conference on Computer Vision and Pattern Recognition*. pp. 1018–1025.

Matas, J., O. Chum, M. Urban, and T. Pajdla: 2002, 'Robust Wide Baseline Stereo from Maximally Stable Extremal Regions'. In: *British Machine Vision Conference*, Vol. I. pp. 384–393.

Mikolajczyk, K. and C. Schmid: 2001, 'Indexing based on scale invariant interest points'. In: *International Conference on Computer Vision*. Vancouver, Canada, pp. 525–531.

Mikolajczyk, K. and C. Schmid: 2002, 'An affine invariant interest point detector'. In: *European Conference on Computer Vision*, Vol. I. pp. 128–142.

Mikolajczyk, K. and C. Schmid: 2003, 'A performance evaluation of local descriptors'. In: *Conference on Computer Vision and Pattern Recognition*.

Moreels, P., M. Maire, and P. Perona: 2004, 'Recognition by Probabilistic Hypothesis Construction'. In: *European Conference on Computer Vision*.

Mundy, J. L. and A. Zisserman: 1992, *Geometric Invariance in Computer Vision*. MIT Press.

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

Murase, H. and S. K. Nayar: 1995, 'Visual Learning and Recognition of 3-D Objects from Appearance'. *International Journal of Computer Vision* **14**, 5–24.

Nalwa, V. S.: 1988, 'Line-drawing interpretation: A mathematical framework'. *International Journal of Computer Vision* **2**, 103–124.

Pentland, A., B. Moghaddam, and T. Starner: 1994, 'View-Based and Modular Eigenspaces for Face Recognition'. In: *Conference on Computer Vision and Pattern Recognition*. Seattle, WA.

Poelman, C. J. and T. Kanade: 1997, 'A Paraperspective Factorization Method for Shape and Motion Recovery'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(3), 206–218.

Ponce, J.: 2000, 'On Computing Metric Upgrades of Projective Reconstructions Under the Rectangular Pixel Assumption'. In: *Second SMILE Workshop*. pp. 18–27.

Ponce, J., D. Chelberg, and W. Mann: 1989, 'Invariant properties of straight homogeneous generalized cylinders and their contours'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**(9), 951–966.

Pope, A. R. and D. G. Lowe: 2000, 'Probabilistic Models of Appearance for 3-D Object Recognition'. *International Journal of Computer Vision* **40**(2), 149–167.

Pritchett, P. and A. Zisserman: 1998, 'Wide Baseline Stereo Matching'. In: *International Conference on Computer Vision*. Bombay, India, pp. 754–760.

Rothganger, F., S. Lazebnik, C. Schmid, and J. Ponce: 2003, '3D Object Modeling and Recognition Using Affine-Invariant Patches and Multi-View Spatial Constraints'. In: *Conference on Computer Vision and Pattern Recognition*, Vol. II. pp. 272–277.

Rothganger, F., S. Lazebnik, C. Schmid, and J. Ponce: 2004, 'Segmenting, Modeling, and Matching Video Clips Containing Multiple Moving Objects'. In: *Conference on Computer Vision and Pattern Recognition*. In press.

Schaffalitzky, F. and A. Zisserman: 2002, 'Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?"'. In: *European Conference on Computer Vision*, Vol. I. pp. 414–431.

Schmid, C. and R. Mohr: 1997, 'Local Grayvalue Invariants for Image Retrieval'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**(5).

Schneiderman, H. and T. Kanade: 2000, 'A Statistical Method for 3D Object Detection Applied to Faces and Cars'. In: *Conference on Computer Vision and Pattern Recognition*.

Selinger, A. and R. Nelson: 1999, 'A Perceptual Grouping Hierarchy for Appearance-Based 3D Object Recognition'. *Computer Vision and Image Understanding* **76**(1), 83–92.

Tell, D. and S. Carlsson: 2000, 'Wide Baseline Point Matching Using Affine Invariants Computed from Intensity Profiles'. In: *Proc 6th ECCV*. Dublin, Ireland, pp. 814–828, Springer LNCS 1842-1843.

Thompson, D. and J. Mundy: 1987, 'Three-dimensional model matching from an unconstrained viewpoint'. In: *International Conference on Robotics and Automation*. Raleigh, NC, pp. 208–220.

Tomasi, C. and T. Kanade: 1992, 'Shape and Motion from Image Streams: a Factorization Method'. *International Journal of Computer Vision* **9**(2), 137–154.

Torr, P. and A. Zisserman: 2000, 'MLESAC: A New Robust Estimator with Application to Estimating Image Geometry'. *Computer Vision and Image Understanding* **78**(1), 138–156.

Triggs, B., P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon: 1999, 'Bundle Adjustment - A Modern Synthesis'. In: B. Triggs, A. Zisserman, and R. Szeliski (eds.): *Vision Algorithms*. Corfu, Greece, pp. 298–372, Spinger-Verlag. LNCS 1883.

Turk, M. and A. Pentland: 1991, 'Eigenfaces for Recognition'. *Journal of Cognitive Neuroscience* **3**(1), 71–86.

Tuytelaars, T. and L. Van Gool: 2004, 'Matching Widely Separated Views based on Affinely Invariant Neighbourhoods'. *International Journal of Computer Vision*. In press.

Voorhees, H. and T. Poggio: 87, 'Detecting Textons And Texture Boundaries In Natural Images'. In: *International Conference on Computer Vision*. pp. 250–258.

Weber, M., M. Welling, and P. Perona: 2000, 'Unsupervised Learning of Models for Recognition'. In: *European Conference on Computer Vision*.

Weinshall, D. and C. Tomasi: 1995, 'Linear and Incremental Acquisition of Invariant Shape Models from Image Sequences'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**(5), 512–517.