A NOVEL APPROACH TO MODELING 3D OBJECTS FROM STEREO VIEWS AND RECOGNIZING THEM IN PHOTOGRAPHS

BY

AKASH M. KUSHAL

B.Tech, Indian Institute of Technology, Delhi, 2003

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2005

Urbana, Illinois

Abstract

Local appearance models in the neighborhood of salient image features, together with local and/or global geometric constraints, serve as the basis for several recent and effective approaches to 3D object recognition from photographs. However, these techniques typically either fail to explicitly account for the strong geometric constraints associated with multiple images of the same 3D object, or require a large set of training images with much overlap to construct relatively sparse object models. This thesis proposes a simple new method for automatically constructing 3D object models consisting of *dense* assemblies of small surface patches and affine-invariant descriptions of the corresponding texture patterns from *a few* (7 to 12) stereo pairs. Similar constraints are used to effectively identify instances of these models in highly cluttered photographs taken from arbitrary and unknown viewpoints. Experiments with a dataset consisting of 80 test images of 9 objects, including comparisons with a number of baseline algorithms, demonstrate the promise of the proposed approach.

Acknowledgments

I would like to thank my advisor Prof. Jean Ponce for his guidance, advice and patience. He has given me freedom to pursue my research interests and provided a wonderful environment for research. I am very thankful to Dr. Fred Rothganger for helping me get started on the work in this thesis. I would also like to thank my friends Soumyadeb Mitra and Sourabh Bhattacharya for their help during the numerous photography sessions in the lab.

This research was supported in part by the National Science Foundation under grants IIS 03-12438 and IIS 03-08087, the UIUC-Toyota collaboration for 3D object modeling, recognition and classification from photographs, and the Beckman Institute.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This work addresses the problem of recognizing three dimensional (3D) objects in photographs taken from arbitrary viewpoints. Recently, object recognition approaches based on local viewpoint invariant feature matching ([16], [8], [10], [9]) have become increasingly popular. The local nature of these features provides tolerance to occlusions and their viewpoint invariance provides tolerance to changes in object pose. Most methods (for example [7],[3]) match each of the training images of the object to the test image independently and use the highest matching score to detect the presence/absence of the object in the test image. This essentially reduces object recognition to a wide-baseline stereo matching problem. Only a few previous approaches ([8], [2], [14]) exploit the relationships among the model views. Lowe [8] clusters the training images into model views and links matching features in adjacent clusters. Each test image feature matched to some feature f in a model view v votes for v and its neighbors linked to f. This helps to model feature appearance variation since different model views provide slightly different pictures of the features they share, yet features' votes do not get dispersed among competing model views. Ferrari *et al.* [2] integrate the information contained in successive images by constructing *region tracks* consisting of the same region of the object seen in multiple views. They introduce the notion of a group of aggregated matches (*GAM*) which is a collection of matched regions on the same surface of the object. The region tracks are then used to transfer matched GAMs from one model view to another, and their consistency is checked using a heuristic test. The problem with this (as with all other methods that do not explicitly exploit 3D constraints) is that geometric consistency can only be *loosely* enforced. Also, for both [8] and [2] there is no way to determine consistency among matched regions which are not seen together in any model view. Rothganger *et al.* [14]

use multiple images to build a model encoding the 3D structure of the object, and the much tighter constraints associated with the 3D projection of the model patches are used to guide matching during recognition. In this case, the 3D model explicitly integrates the various model views, but the determination of the 3D position and orientation of a patch on the object requires it to be visible in three or more training images [14], and hence requires a large number of closely separated training images for modeling the object. Also, [14] only makes use of patches centered at interest points, so the model constructed is sparse and does not encode all the available information in the training images. We tackle these issues by using calibrated stereo pairs to construct partial 3D object models and then register these models together to form a full model.[1] This allows the use of a sparse set of stereo training views (7 to 12 pairs in our experiments) for the modeling. We also extend to 3D object models the idea proposed in [3] in the image matching domain, and augment the model patches associated with interest points of [14] (called *primary* patches from now on) with more general *secondary* patches. This allows us to cover the object densely, utilize all the available texture information in the training images, and effectively handle clutter and occlusion in recognition tasks.

1.1 Overview of Our Approach

Our approach consists of three key steps.

1. Detection and description of affine invariant interest points (*affine regions*) that provide a normalized, viewpoint independent description of local image appearance.

2. Calibrated stereo matching and 3D reconstruction of the primary and secondary patches on the left and right images of the training stereo pairs to construct partial 3D models. Combining the partial 3D models for the different stereo views into full 3D models of the objects.

3. Employing both photometric and geometric consistency constraints to match groups of

[1] This is for modeling only of course; individual photographs are used for recognition.

patches during recognition.

We follow a scheme similar to [15] for the detection and description of affine regions. Chapter 2 (adapted from [15]) provides the necessary background and specific details of the implementation.

As was mentioned previously, we use a set of calibrated stereo views for determining the 3D structure and building a model of the object. Potential *primary matches* between the affine regions found in each stereo pair are first filtered using photometric and geometric consistency constraints, and then augmented with additional *secondary matches* for a dense coverage of the object, as proposed in [3] for the 2D case. The 3D location and shape of the patches is determined using a standard stereo algorithm to generate partial models which are later combined to form a complete model of the object. The 3D patches that correspond to primary (or secondary) matches are called primary (or secondary) model patches.

A similar scheme is followed during recognition. First, the primary patches in the model are matched to the affine regions found in the test image. These primary patches are then used as guides for matching nearby secondary patches. The recognition decision is based on the number of matched patches.

The thesis is organized as follows. Chapter 2 describes the detection and representation of affine invariant patches. The construction of the partial models and their inter-registration to generate the full model is explained in chapter 3. The details of the recognition algorithm are provided in chapter 4. In chapter 5 we show recognition results using the proposed approach and discuss future extensions of this work.

Chapter 2

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. This chapter presents the approach to the detection and description of *affine regions* used in our implementation. Most of the material in this chapter is adapted from [15].

2.1 Detection

Our work uses a form of the affine-covariant region detector developed by Mikolajczyk and Schmid [10]. This algorithm depends on a separate interest point detector to provide a set of points along with their initial scales. A study by Mikolajczyk et al. [11] concludes that no single detector outperforms the others on all types of scenes and image transformations. Therefore, in the absence of prior knowledge about the type of scene, it is beneficial to use a battery of complementary detectors. The primary detectors we use are the Harris-Laplacian detector and the difference-of-Gaussians (DoG) operator [1, 6, 17]. The Harris detector tends to find corners and points at which significant intensity changes occur (considered to be regions of "high information content" [10]) while the DoG detector is in general attracted to the centers of roughly uniform regions (blobs).

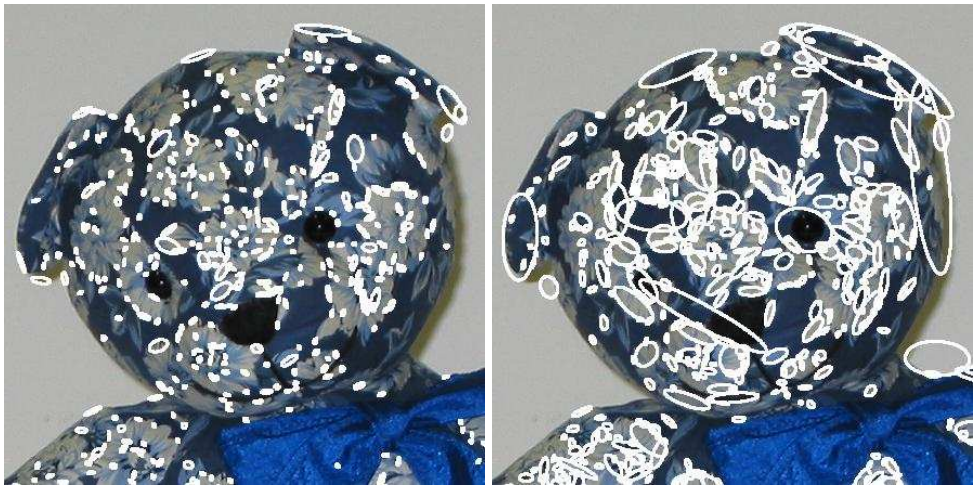


Figure 2.1: (After [15].) Affine-adapted patches found by the Harris-Laplacian (left) and the DoG (right) detector.

Figure 2.1 shows examples of the outputs of these two detectors. We modify the affine adaptation procedure proposed by Mikolajczyk and Schmid by also computing an orientation for each patch. The standard output of affine adaptation are elliptical-shaped patches. 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, turning the corresponding ellipse into a parallelogram and the unit circle into a square (Figure 2.2). Thus, the output of the detection process is a set of image regions in the shape of parallelograms. Each parallelogram shaped patch is completely defined by the rectifying transformation R that maps the parallelogram onto a "unit" square centered at the origin or equivalently by the inverse rectification transformation $S = R^{-1}$ that maps the rectified unit square into the parallelogram in the image (Figure 2.3(b)).

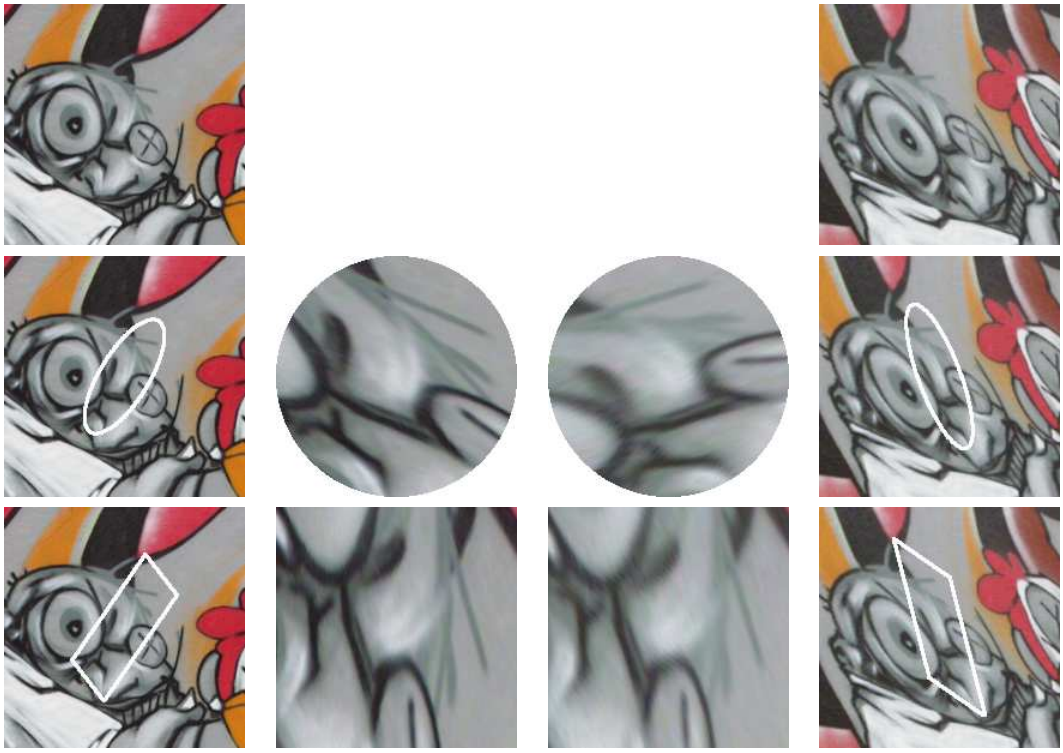
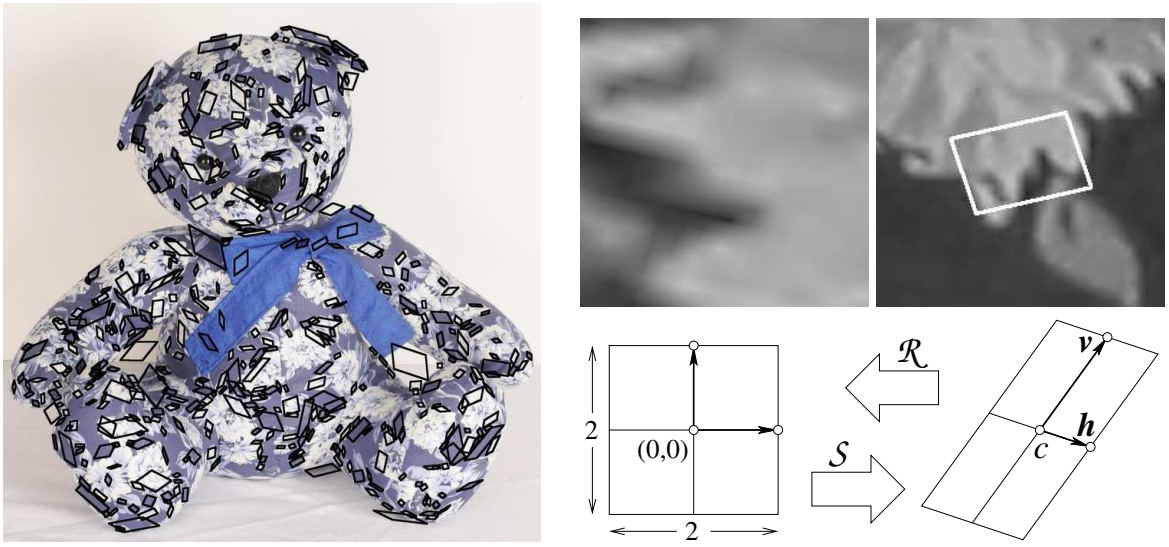


Figure 2.2: (After [15]) Normalizing patches. The left two columns show a patch from image 1 of Krystian Mikolajczyk's graffiti dataset. The right two columns show the matching patch from image 4. The first row shows the region of the original image. The second row shows the ellipse determined by affine 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.

2.2 Description

A rectified affine region is a *normalized* representation of the local surface appearance, invariant under planar affine transformations. Under affine (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.



(a) Affine regions found in an image of a teddy bear. Only a subset of the patches detected is shown for clarity.

(b) (After [15]) The inverse transformation S maps the rectified square associated with an affine region back onto the image.

Figure 2.3: Affine regions and inverse rectification.

Other feature spaces may of course be used as well. In particular, the SIFT descriptor introduced by Lowe [7] has been shown to provide superior performance in image retrieval tasks [12]. 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 quantized into 8 bins (Figure 2.4), and it can thus be represented by a 128-dimensional feature vector [6]. Following [15] we combine the SIFT feature vector with a color histogram in the YUV color space. The histogram is two-dimensional (typically 10×10) and built only from the chroma component, that is, the U and V values. Figure 2.4 shows an example of the color histogram.

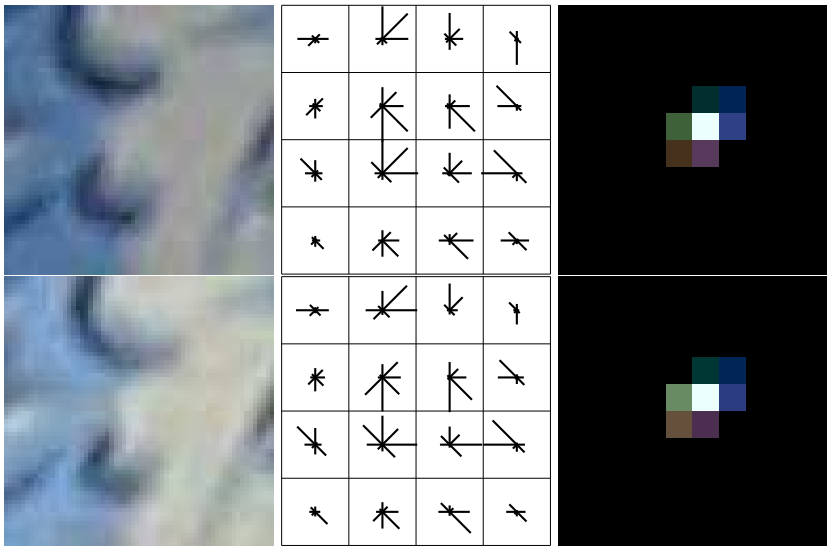


Figure 2.4: (After [15]) Two (rectified) matching patches found in two images of a teddy bear, along with the corresponding SIFT and color descriptors. The orientation histogram values associated with each spatial bin are depicted by lines of different lengths for each one of the 8 quantized gradient orientations. As recommended in [6], we scale the feature vectors associated with SIFT descriptors to unit norm, and compare them using the Euclidean distance.

Chapter 3

Stereo Modeling

We start by acquiring a few (7 to 12) stereo pairs that are roughly equally spaced around the equatorial ring of the object for modeling. The stereo views are taken against a uniform background to allow for easy segmentation. Then, a standard stereo matching algorithm that searches for matching patches along corresponding epipolar lines is used to determine an initial set of tentative matches. We use a combination of SIFT [7] and the color histogram descriptor described in [15] to compute the initial matches. The matches are then refined to obtain the correct alignment of the patches in the left and right images. Only matches with normalized correlation greater than a pre-refinement threshold (kept at 0.75) are considered for the refinement step for efficiency reasons. The refinement process employs nonlinear optimization to affinely deform the right image patch until the correlation with its match in the left image is maximized. Matches with normalized correlation greater than a post-refinement threshold (equal to 0.9 for this work) are kept for subsequent processing.

The matches are filtered by using a neighborhood constraint which removes a match if its neighbors are not consistent with it. More precisely, for every match m we look at its K closest neighbors in the left image ($K = 5$ in our implementation) and, for every triple out of these, we calculate the barycentric coordinates of the center of the left and right patches of m with respect to the triangle formed by the centers of the patches of the triple in the left and right images respectively. We then count the number of triples for which these barycentric coordinates agree (the sum of squared differences is smaller than a tolerance limit $\mathcal{L} = 0.5$). We repeat the process using the K closest neighbors of m in the right image and add up both the counts. Finally, the matches with a count smaller than a threshold T are dropped. Setting $T = 2\binom{K-1}{3}$ ensures that a

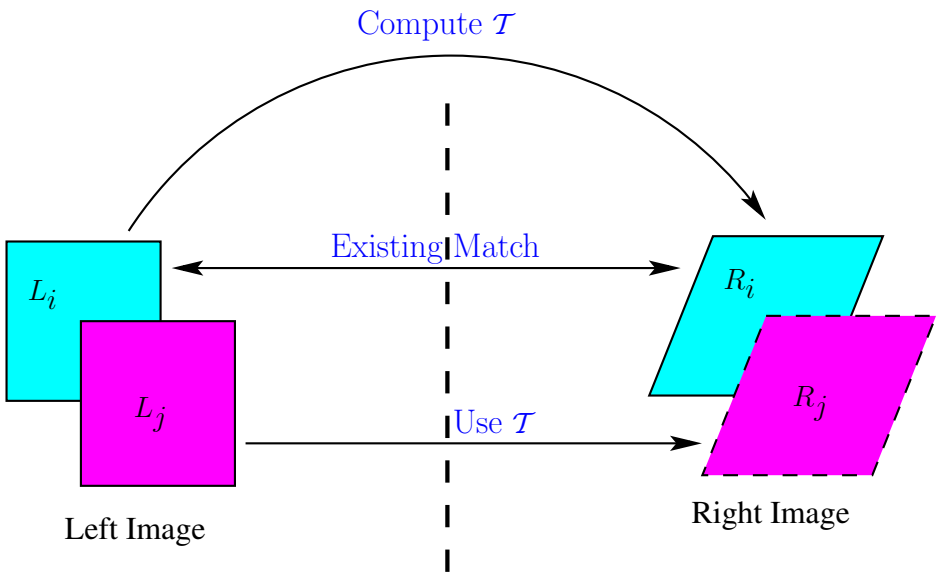


Figure 3.1: Expansion during initial matching.

correct match with one bad nearby match out of the K still survives after this test. This gives us a set Γ of reliable matches. Note that these matches are based only on the primary patches associated with salient affine regions detected in the stereo training images and hence, only cover the object sparsely. To get a dense coverage of the object we use an expansion technique similar to [3] to spread these initial matches in Γ.

3.1 Expansion Technique

We use the fact that the training views are taken against a uniform background to segment the object and cover it with a grid Ω of partially overlapping square-shaped patches in the left image (Fig. 3.2(a)). For every match m_i in Γ, we compute the affine transformation $\mathcal{T} = \mathcal{S}_{R_i}\mathcal{S}_{L_i}^{-1}$ between the corresponding patches L_i and R_i in the left and right images. Here $\mathcal{S}_{L_i}$ and $\mathcal{S}_{R_i}$ are the inverse rectification matrices for L_i and R_i respectively. We use $\mathcal{T}$ to predict the location $\mathcal{S}_{R_j} = \mathcal{T}\mathcal{S}_{L_j}$ of the right matches of the yet unmatched patches L_j in Ω that are close to (within one side length of) the center of L_i. This process is shown diagramatically in figure 3.1. Then, a refinement process is used to align the predicted patch correctly in the right image. Again, if the match has sufficient

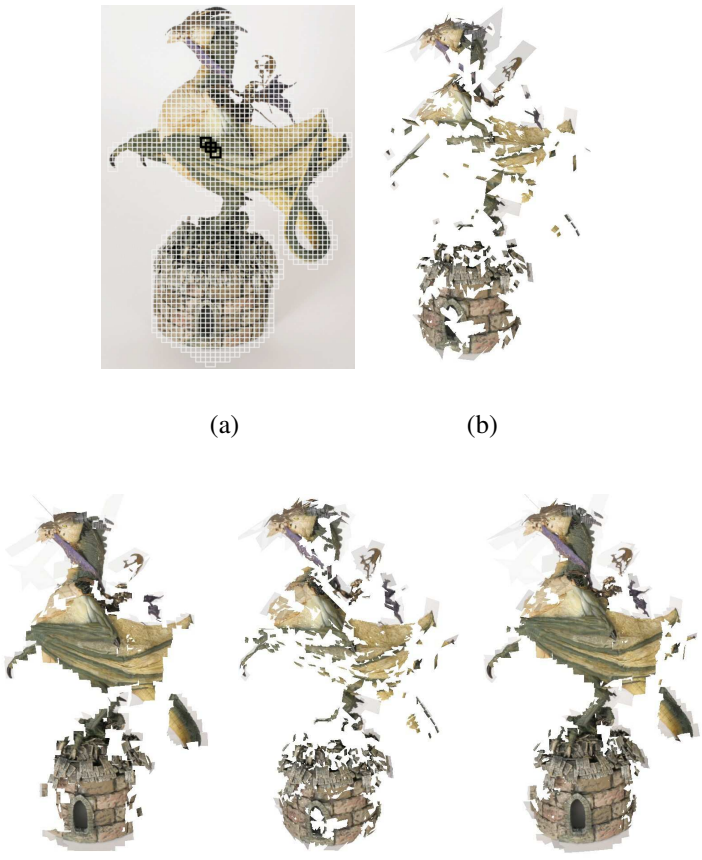
correlation after refinement, it is accepted as a valid match and added to Γ. Since the patches that form these matches are not associated with interest points, we call these secondary matches. The expansion process iterates by expanding around the newly added matches to Γ until no more matches can be added. This process usually covers the entire object surface densely with matches. Figure 3.2(c) shows the secondary patches on a partial model of the dragon constructed from a single stereo pair.

We then use the secondary matches to locate additional primary matches associated with salient affine regions. Even though the corresponding part of the object surface may already be covered (with secondary matches), this is useful because it is the primary matches that can be repeatably detected, and will later be required for the initial matching to the test image as well as for the alignment of the partial models. This is accomplished by finding unmatched affine regions in the left (respectively right) image, and using close-by secondary matches to predict the position of the corresponding patches in the right (respectively left) image. Again, a refinement process is used to adjust the alignment of the right (respectively left) image patch. If there is sufficient correlation (again 0.9) between the left and right patches, the match is added to Γ. Figures 3.2(d) and 3.2(e) respectively show the expanded primary patches and the union of the primary and secondary patches in the partial model of the dragon.

3.2 Model Construction

The dense matches constructed as discussed above are used for building partial 3D models (one for each stereo pair). First, we solve for the patch centers in 3D by using standard calibrated triangulation techniques. Then we reconstruct the edges of the corresponding parallelograms using a first-order approximation to the perspective projection equations in the vicinity of the patch centers as proposed by Rothganger [15]. We provide a brief sketch (adapted from [15]) of the algorithm below.

Consider the homogeneous projection equation



(a) (b)

(c) (d) (e)

Figure 3.2: (a) Left image in a stereo pair, covered with a grid of patches (three of the overlapping patches are shown in black for clarity). (b) Partial model constructed from primary matches before expansion. (c) Model constructed using only the secondary patches found during expansion. (d) Model containing the primary patches after expansion. (e) Model containing all the patches after expansion.

$$\begin{bmatrix} \boldsymbol{p} \\ 1 \end{bmatrix} = \frac{1}{z}\mathcal{M}\begin{bmatrix} \boldsymbol{P} \\ 1 \end{bmatrix}, \quad \text{where} \quad \mathcal{M} = \begin{bmatrix} \mathcal{A} & \boldsymbol{b} \\ \boldsymbol{a}_3^T & 1 \end{bmatrix}$$

is the perspective projection matrix, $\mathcal{A}$ is a 2×3 sub-matrix of $\mathcal{M}$, $\boldsymbol{p}$ is the non-homogeneous coordinate vector for the point in the image, and $\boldsymbol{P}$ is the non-homogeneous coordinate vector of

the point in 3D. We can write the perspective projection mapping as

$$\boldsymbol{p} = f(\boldsymbol{P}) = \frac{1}{\boldsymbol{a}_3 \cdot \boldsymbol{P} + 1}(\mathcal{A}\boldsymbol{P} + \boldsymbol{b}),$$

and a Taylor expansion of order 1 of the function f in $\boldsymbol{P}$ yields $f(\boldsymbol{P} + \boldsymbol{\delta P}) = \boldsymbol{p} + \boldsymbol{\delta p} = f(\boldsymbol{P}) + f'(\boldsymbol{P})\boldsymbol{\delta P}$, or

$$\begin{aligned}
\boldsymbol{\delta p} &= f'(\boldsymbol{P})\boldsymbol{\delta P} \\
&= \frac{\mathcal{A}(\boldsymbol{a}_3 \cdot \boldsymbol{P} + 1) - (\mathcal{A}\boldsymbol{P} + \boldsymbol{b})\boldsymbol{a}_3^T}{(\boldsymbol{a}_3 \cdot \boldsymbol{P} + 1)^2}\boldsymbol{\delta P} \\
&= \frac{1}{\boldsymbol{a}_3 \cdot \boldsymbol{P} + 1}(\mathcal{A} - \boldsymbol{p}\boldsymbol{a}_3^T)\boldsymbol{\delta P}.
\end{aligned}$$

The basis vectors $\boldsymbol{H}$ and $\boldsymbol{V}$ of the 3D patch are essentially small changes around the patch center $\boldsymbol{C}$, so they play the role of $\boldsymbol{\delta P}$. Let $\boldsymbol{h}$ and $\boldsymbol{v}$ be the projections of $\boldsymbol{H}$ and $\boldsymbol{V}$ into the image. The linearized projection equations for the patch can be written as follows.

$$\begin{aligned}
\boldsymbol{h} &= f'(\boldsymbol{C})\boldsymbol{H}, \\
\boldsymbol{v} &= f'(\boldsymbol{C})\boldsymbol{V}
\end{aligned}$$

We stack up 4 equations (2 for the left and 2 for the right camera of the stereo pair) for each of H and V and solve them using linear least squares to obtain the basis vectors and hence determine the 3D location of the parallelogram patch. Doing this for all the matches gives us a partial 3D model of the object for each stereo pair. The next task is to combine these partial models into a complete model.

3.3 Registration of Partial Models

Algorithm 1 gives a concise description of the steps involved in registering the partial models

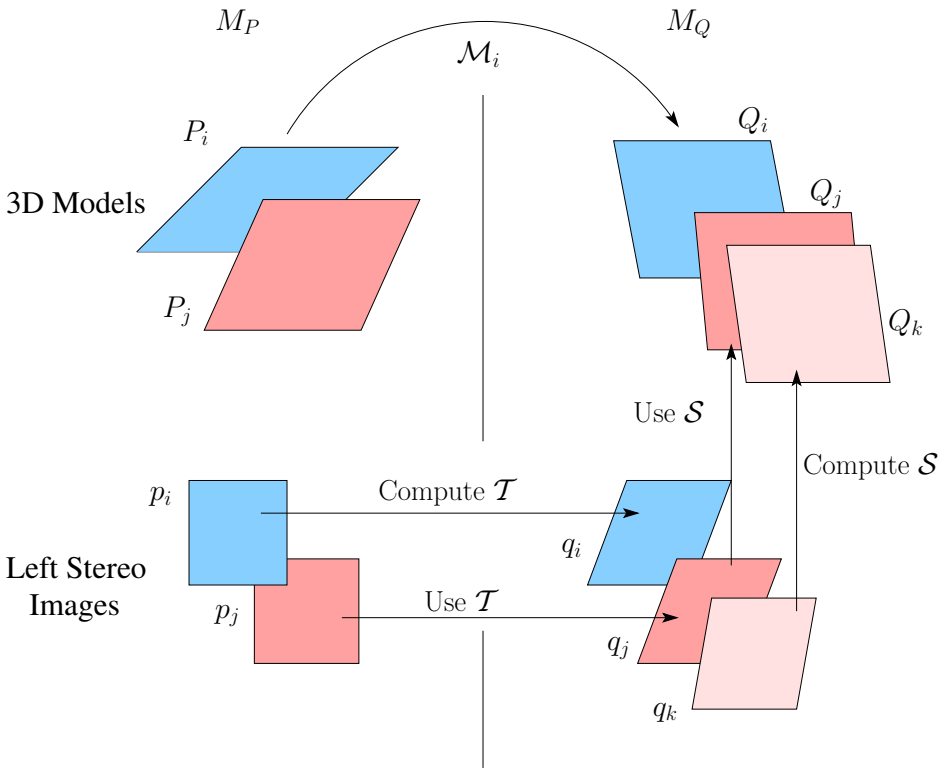


Figure 3.3: Expansion during registration.

together. The first step in combining the models is to find appearance-based matches between the primary model patches in adjacent partial models. Again, SIFT and color histogram descriptors are used to facilitate the initial matching.

3.3.1 Match Expansion

A variant of the 2D expansion scheme described earlier is used to propagate these initial matches between 3D patches to neighboring model patches as follows (Fig. 3.3). Let the two partial models being registered be M_P and M_Q. For each initial match $\mathcal{M}_i$ between the 3D patches $\mathcal{P}_i$ in M_P and $\mathcal{Q}_i$ in M_Q, we consider the 2D patch p_i (resp. q_i) corresponding to $\mathcal{P}_i$ (resp. $\mathcal{Q}_i$) in the left stereo image of M_P (resp. M_Q). We calculate the affine transformation $\mathcal{T}$ that maps the patch p_i onto q_i. Then, we consider the yet unmatched patches P_j in M_P whose 2D projection p_j in the left stereo image lies within a small distance limit of the center of p_i. These patches p_j are then projected to q_j in the left stereo image of M_Q using $\mathcal{T}$. A non-linear match refinement process (similar to the

Input: A set of partial models $\mathcal{S}_M = \{\mathcal{M}_1, \ldots \mathcal{M}_K\}$.
Output: A combined model $\mathcal{M}$.

for all pairs of consecutive partial models $\mathcal{M}_i, \mathcal{M}_j \in \mathcal{S}_M$ **do**

Step 1: Appearance based selection of potential matches

• Use SIFT and color-histogram descriptors to match the primary patches between $\mathcal{M}_i$ and $\mathcal{M}_j$ to produce a set $\mathcal{T}_{ij}$ of tentative matches.

• Use the non-linear match refinement process to update the match parameters to optimize the normalized correlation. Remove matches with normalized correlation $< \tau$ from $\mathcal{T}_{ij}$.

Step 2: Match expansion

• Expand the matches $\mathcal{T}_{ij}$ using the method described in section 3.3.1

Step 3: RANSAC

• Use RANSAC to robustly estimate the rigid transformation $\mathcal{R}_{ij}$ between $\mathcal{M}_i$ and $\mathcal{M}_j$ and determine a large subset $\mathcal{S}_{ij} \subset \mathcal{T}_{ij}$ consistent with $\mathcal{R}_{ij}$.

end for

Step 4: Refinement

• Use $\mathcal{R}_{ij}$'s initialize the position and orientation $\mathcal{P}_{\mathcal{M}_i}$ of all the partial models $\mathcal{M}_i$ in the coordinate frame attached to the first partial model.

repeat

for all partial models $\mathcal{M}_i$ **do**

• Let the neighboring partial models of $\mathcal{M}_i$ be $\mathcal{M}_j$ and $\mathcal{M}_k$.

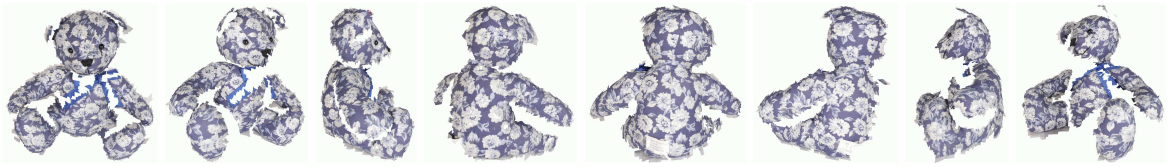
• Update the position $\mathcal{P}_{\mathcal{M}_i}$ of $\mathcal{M}_i$ so as to minimize the sum of squared errors between the centers of the matched patches for all the matches in $\mathcal{S}_{ij}$ and $\mathcal{S}_{ik}$ using the algorithm of section 3.3.3

end for

until convergence

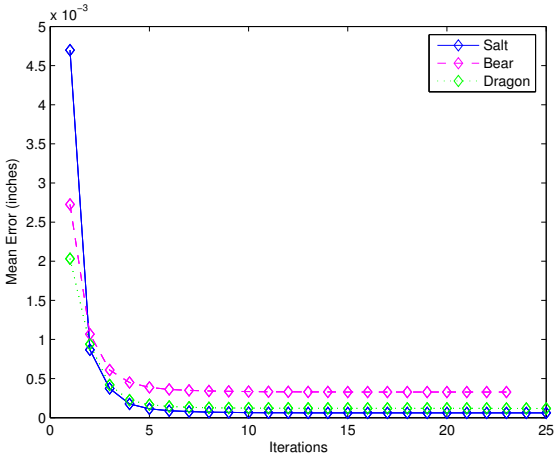
Algorithm 1: Registration of partial models.

one described earlier) is then used to align the projected patch q_j correctly. The match is removed from consideration if the final correlation between p_j and q_j's normalized representation is less than a threshold (again kept at 0.9). If the match passes this test we find the patch Q_k in M_Q whose projection q_k into the left stereo image of M_Q is closest to q_j's center point. An estimate of the position of the 3D patch Q_j that corresponds to the 2D patch q_j can then be obtained, assuming that Q_j lies on the same plane π_k as Q_k. An affine transformation $\mathcal{S}$ that maps the 2D patch q_k to the 3D patch Q_k on π_k is calculated and then Q_j is estimated by projecting q_j onto π_k using $\mathcal{S}$. This new match between P_j and Q_j is then added to the set of matches and is used for finding other matches. This expansion step has proven to be very useful while registering models with small overlap.



(a) Partial models

(b) Complete model



(c) Mean Error during Refinement

Figure 3.4: Registration of partial models.

3.3.2 RANSAC

All the matches generated above are filtered through a classical RANSAC procedure that finds the matches consistent with a rigid transformation. RANSAC [4] is a *robust estimation* algorithm that considers 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. Briefly, RANSAC iterates over two steps: In the *sampling* stage, a (usually, but not always) minimal set of matches is chosen randomly, and this "seed" set 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.

In our particular case we aim to estimate the rigid transformation that best aligns the two consecutive partial models. In each iteration a set of 3 matches are randomly chosen and used to estimate the rigid transformation that minimizes the sum of squared distances among the matched patch centers after alignment. The matches consistent with this transformation are collected to form a consensus set and the largest consensus set in all the iterations is finally used to estimate the parameters of the rigid transformation.

3.3.3 Refinement

The above RANSAC procedure provides an estimate of the pairwise rigid transformations. Since these pairwise estimates may not in general be consistent with each other (the product of the rotations between the consecutive models must be the identity), we use a process similar to [13] to find a consistent solution: It is initialized using the pairwise transformation estimates and these estimates are refined by looping through all the partial models and updating the position of the current model to align it best with its neighbors. More formally, we search for the rigid transformation that minimizes the sum of squared distances between the centers of the matched patches in the current model and its neighbors. The positions of these neighbors are kept fixed while the position of the current partial model is calculated via linear least squares using quaternions [5]. In the following, we briefly describe the mathematics involved (adapted from [5]).

Note that when we modify the position of some partial model $\mathcal{M}$, the other partial models are kept fixed and hence the problem is one of finding a rigid transformation R, t that minimizes

$$E = \sum_{i=1}^{n} |x'_i - Rx_i - t|^2$$

where x_i are the center points of the parallelogram patches of $\mathcal{M}$ and x'_i are the center points of the matching patches in neighboring models of $\mathcal{M}$. The value of t minimizing E can be obtained

by setting the partial derivative $\frac{\partial E}{\partial t} = 0$ which gives,

$$t = \bar{x}' - R\bar{x} \quad \text{where} \quad \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \quad \text{and} \quad \bar{x'} = \frac{1}{n}\sum_{i=1}^{n} x'_i$$

We can remove t from the minimization by changing variables to centered points $y_i = x_i - \bar{x}$ and $y'_i = x'_i - \bar{x'}$. This gives us,

$$E = \sum_{i=1}^{n} |y'_i - Ry_i|^2$$

Let q denote the quaternion associated with the matrix R. See [5] for an explaination of quaterions and their properties. Using $|q|^2 = 1$ we can write

$$E = \sum_{i=1}^{n} |y'_i - qy_i\bar{q}|^2|q|^2 = \sum_{i=1}^{n} |y'_i q - qy_i|^2$$

where $\bar{q}$ is the conjugate quaternion of q. If we represent q by a 4-vector whose first element is the real part of q and last 3 elements are the imaginary part we can rewrite $E = q^T\mathcal{B}q$ where $\mathcal{B} = \sum_{i=1}^{n} \mathcal{A}_i^T\mathcal{A}_i$ and

$$\mathcal{A}_i = \begin{bmatrix} 0 & y_i^T - y_i'^T \\ y'_i - y_i & [y_i + y'_i]_\times \end{bmatrix}$$

Minimizing E in this form under the constraint that $|q|^2 = 1$ is now easy and the optimal q is just the eigenvector of $\mathcal{B}$ corresponding to its smallest eigenvalue.

The above process is iterated until the sum of squared errors between all the matches between all the pairs of consectutive partial models converges to some local minimum. Figure 3.4(c) shows a plot of the mean squared error after each iteration of the refinement process for three of the models used for experimentation. Finally the rigid transformations estimated are used to bring all the partial models into a common euclidean coordinate frame and a complete model is constructed by taking the union of these transformed partial models. The partial models and the complete model formed after registration for a teddy bear are shown in Figs. 3.4(a) and 3.4(b) respectively.

Chapter 4

Recognition

The first part of the recognition process is similar to [14] in which the repeatable primary patches in the 3D model are matched to the interest points detected in the test image and the matches with high appearance similarity are selected. Again, we use both SIFT descriptors and color histograms to characterize the appearance of the patches and compute the initial matches. The refinement process is then employed to maximize the correlation between the matched test image patch and the corresponding model patch. Matches that have sufficient correlation (again taken as 0.9) after the refinement step are accepted and the others are dropped before further processing. These matches are then used as seeds for the subsequent match expansion stage. Algorithm 2 provides a summary of the entire recognition algorithm.

4.1 Expansion Process

This process is similar in spirit to the expansion technique used during the initial modeling but the expansion here happens on the surface of the 3D model instead of the stereo images. For this, we first preprocess the model M to build an undirected graph G_M that represents the adjacency information of the patches in M. We add an edge e between two patches if their centers lie within a distance limit of each other. This limit is set to be such that the average degree of a vertex is around 20. We now spread the matches along the edges of this graph using the following expansion steps.

- **Expansion using images (Fig. 4.1(a)):** This step is used at the start when the matches have not been filtered through a geometric consistency check so the test image camera cannot be estimated reliably. This works similar to the modeling case, and for each previously matched

Input: A model $\mathcal{M}$ and a set of affine regions T on the test image.
Output: A set S of trusted matches.

- Match the primary patches in $\mathcal{M}$ to the affine regions in T using SIFT and color-histogram descriptors to produce a set of putative matches P.
- Run non-linear match refinement on the matches in P and keep only those with normalized correlation $\geq \tau$.
- Use the image-based expansion step to add matches to P.

repeat

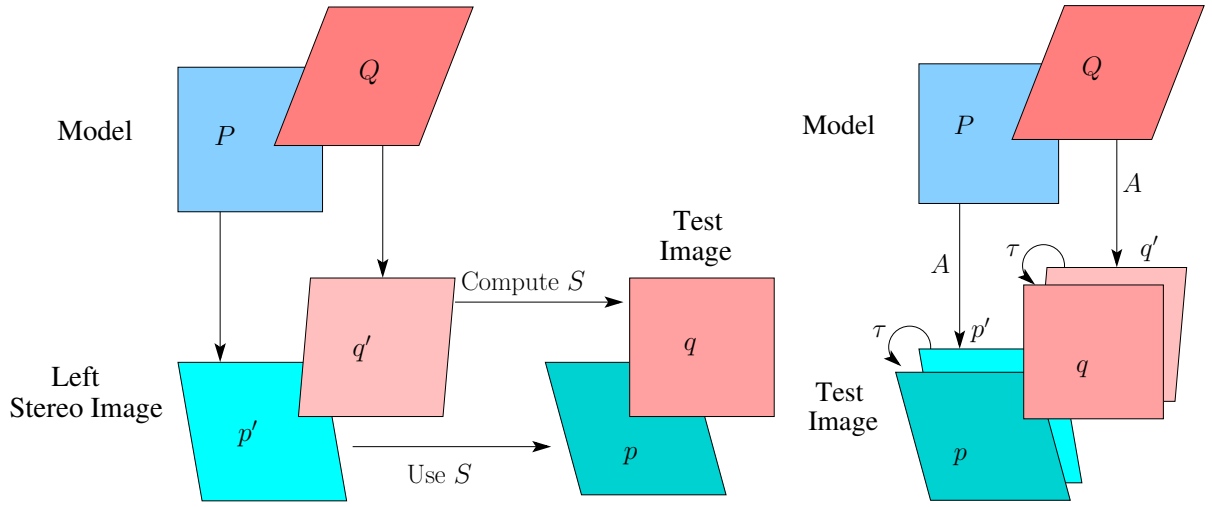
- Run the geometric consistency test described in section 4.2 on P and update P with the set of consistent matches
- Set C to the estimated camera.
- Run the camera-based expansion step using the camera estimate C and add the new matches to P
- Use C to project all the primary patches in $\mathcal{M}$ into the test image and match to nearby affine regions detected in the image. Add the obtained matches to P.

until cardinality of P stops increasing

Algorithm 2: Recognition algorithm

model patch P we calculate the affine transformation S that maps its projection in the left training image of the stereo pair from which it originates into the test image. Then we look at every unmatched neighbor Q of P that is part of the same partial model (and so shares the same left stereo image) and use S to predict its location in the test image. This predicted position is then refined as before and the match is accepted if the correlation is sufficiently large (again compared to 0.9). This expansion scheme does not allow expanding matches from one partial model to another.

- **Expansion using the camera (Fig. 4.1(b)):** This step is used after the matches have been filtered through a geometric consistency check and the camera A associated with the test image has been estimated. A is used to project a base 3D patch P (which is already matched to a patch p in the test image) and some adjacent patch Q into the test image. Let the 2D projected patches be p' and q' respectively. A correcting affine transformation τ is computed that aligns the projection p' of the base 3D match exactly with its correct location p. τ is then applied to the projection q' of the adjacent patch to obtain a corrected prediction q of



(a) Using images.

(b) Using the camera.

Figure 4.1: Expansion during recognition.

its position. The prediction is then refined as before to maximize the normalized correlation between the rectified patches corresponding to the match and accepted only if it has high correlation (greater than 0.9). This expansion step allows for moving smoothly from one partial model to another and hence provides an advantage over the pure 2D expansion technique of [3].

The two expansion steps also allow us to reject false matches by simply removing those that do not have enough support. More precisely, if the expansion step from a base match tries to expand to a large number of neighbors and none of these succeeds in forming an acceptable match, the base match is removed.

4.2 Geometric Consistency Test

A *"greedy"* RANSAC-like algorithm 3 is used to extract a set of geometrically consistent matches. The camera of the test image is approximated by a weak-perspective camera with zero skew and

Input: A set M of possible matches.
Output: A set S of trusted matches, camera for the test image C

for i = 1 to N **do**
- Pick a match $m_i \in M$ at random.
- Select the most compatible match $m_i' \in M \setminus \{m_i\}$ to m_i.
- Initialize $S_i = \{m_i, m_i'\}$ and C_i to the camera estimated using S_i.
- Set $m_{best} \in M \setminus S_i$ to the match with minimum reprojection error $\mathcal{E}_{best}$ using C_i

while $|S_i| < K$ and $\mathcal{E}_{best} < \tau$ **do**
- $S_i \Leftarrow S_i \cup \{m_{best}\}$.
- Update C_i with the camera estimated using S_i
- Set $m_{best} \in M \setminus S_i$ to the match with minimum reprojection error $\mathcal{E}_{best}$ using C_i

end while
- Add all matches $m \in M \setminus S_i$ with reprojection error $\mathcal{E}_m < \tau$ to S_i.

end for
- Set S to the S_i with the largest cardinality.
- Estimate the camera C for the test image using S.

Algorithm 3: Geometric consistency check.

square pixels. The algorithm starts by picking a match m_i at random and searches among all the other matches for the most compatible one, say m_i'. The compatibility is checked by first using the two matches to estimate the camera for the test image and then computing the reprojection error for the two matches. The algorithm then creates a set of matches S_i compatible with this pair as follows: S_i is initialized as $S_i = \{m_i, m_i'\}$. The algorithm greedily adds to S_i the most compatible match (the one with the least reprojection error) out of all the matches not yet included in S_i. This iterative process continues until either the size of S_i exceeds $K = 10$, or the smallest reprojection error itself exceeds a threshold τ. The estimate of the camera is updated after each addition to S_i during these iterations. If the size of S_i reaches K, the current estimate of the camera is used to reproject the 3D patches for all the matches into the test image and those with reprojection error less than τ are added to S_i. The algorithm iterates a fixed number of times, each time picking a random match m_k and computing the set of consistent matches S_k. Finally the set S with the largest size is chosen as the set of consistent matches.

The recognition algorithm starts by using the image-based expansion step to grow the initial appearance based primary matches. Then the geometric consistency check is run to extract con-

sistent matches and estimate the camera for the test image and more matches are added using the camera based expansion step. For extending matches to parts of the object that are not directly connected to the initial matches in the test image (possibly due to occlusion) the reconstructed test camera is used to project unmatched primary patches from the model into the test image. Affine regions detected in the test image close to these projected positions are then matched to the corresponding model patch. Again, if the correlation after refinement is sufficiently high, the match is accepted. The geometric consistency check and the following expansion steps are iterated until the number of matches does not increase any more.

Chapter 5

Results

We have evaluated the proposed method on a dataset consisting of 9 objects and 80 test images. The object models, constructed from 7 to 12 stereo views each, are shown in Fig. 5.1. The objects vary from simple shapes (e.g., the salt container) to quite complex ones (e.g., the two dragons and the chest buster model).

The test images (shown in Figure 5) contain the objects in different orientations and under varying amounts of occlusion and clutter. The total number of occurrences of the objects in the test image dataset is 129 since some images contain more than one object. Figure 5.4(a) shows the ROC plot between the true positive (detection) rate and the false positive rate. To asses the value of the expansion step of our approach, we have simply removed the secondary patches and the extra primary patches added during this stage of modeling from our models, and used these sparse models for recognition (this is similar in spirit to the algorithm proposed by Rothganger et al. [14], but includes the expansion step during the recognition phase which was absent in [14]). The corresponding recognition performance is depicted by the blue ROC curve. Our experiments clearly demonstrate the benefit of using dense models as opposed to sparse ones for our dataset. We have also implemented recognition as wide-baseline stereo matching to assess the power of using explicit 3D constraints as opposed to simple epipolar ones. Each test image is matched to all the 168 training images (both left and right images for each stereo pair) for every object separately, making a total of $168 \times 80 = 13440$ image pairs to be compared. The maximum number of matches corresponding to each object is recorded and used to construct the ROC curve. As expected, our method clearly outperforms this simple baseline approach. The detection rates for zero false positives and the equal error rates for the different methods are shown in Fig. 5.

Method	Detection Rate (zero false positives)	Equal Error Rate
Proposed Approach	86.8%	89.1%
Primary patches only	69.8%	84.9%
Wide Baseline	58.1%	77.1%

Table 5.1: Error rate comparison.

The proposed approach also performs well on the highly complex geometric objects like the dragons and the chest buster model. Figure 5.4(b) shows the comparison of the ROC plots on the dataset restricted to only these 3 models. The variation in appearance of the features due to small viewpoint changes is larger for these models since the surface of the models is not smooth. Because the proposed approach combines the different views of the features together (when the different partial models are merged) its performance is less severely affected on the restricted dataset. On the other hand, the performance of the wide-baseline matching scheme drops by a significant amount.

Our current implementations of the modeling and recognitions algorithms runs quite slowly. The modeling was done on an 3 GHz, Intel Pentium 4 machine with 1 GB of RAM. The construction of the partial models for each stereo pair takes about 15 to 20 minutes. The registration of all the partial models into a full model takes approximately another 1 to 2 hours. Most of the time during modeling process is spent in the non-linear match refinement procedure. The recognition experiments took varying amounts of time based on whether the object being recognized was actually found in the image. In cases when very few (< 10) matches were found between the object model and the test image the program took less than 30 minutes. But, in cases when a large number of matches were found the program could take as long as 3 hours. Since, it would have taken a very long time to run experiments on a single machine the program was run on a cluster of machines to speedup the experimentation process.

Finally, Figures 5.3 and 5.5 give a qualitative illustration of the performance of our algorithm with a gallery of recognition results on some test images which contain the objects under heavy occlusion, viewpoint and scale variation, as well as extensive clutter.

5.1 Conclusions and Summary

We have proposed an approach to efficiently build dense 3D euclidean models of objects from stereo views and use them for recognizing these objects in cluttered photographs taken from arbitrary viewpoints. At this point there are many directions for future work.

- Extending the approach to handle non rigid deformations
- Recognizing objects in a cluttered scene using a pair of calibrated stereo images of the scene.
- Collaboration among different cameras looking at the same scene for recognizing the objects in the scene.

Also, it would be desirable to do a comparison with the native implementations of other state-of-the-art recognition methods such as those proposed by Ferrari et al. [2], Lowe [8], and Rothganger et al. [14].

(a) Bournvita (8 pairs)

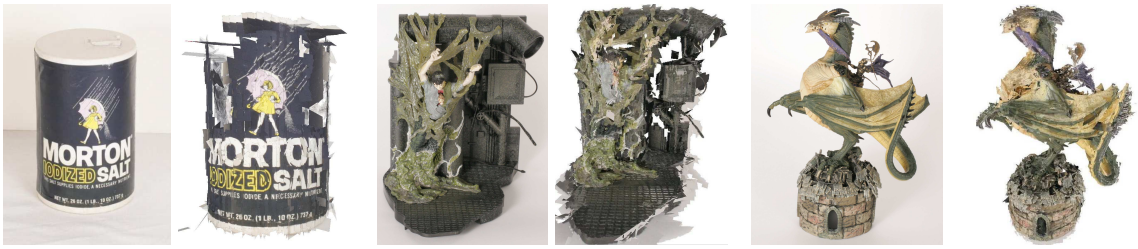
(b) Ball (12 pairs)

(c) Yogurt (8 pairs)

(d) Vase (8 pairs)

(e) Bear (8 pairs)

(f) Small Dragon (12 pairs)



(g) Salt (8 pairs)

(h) Chest Buster (7 pairs)

(i) Dragon (12 pairs)

Figure 5.1: Object models.

Figure 5.2: The test image dataset.

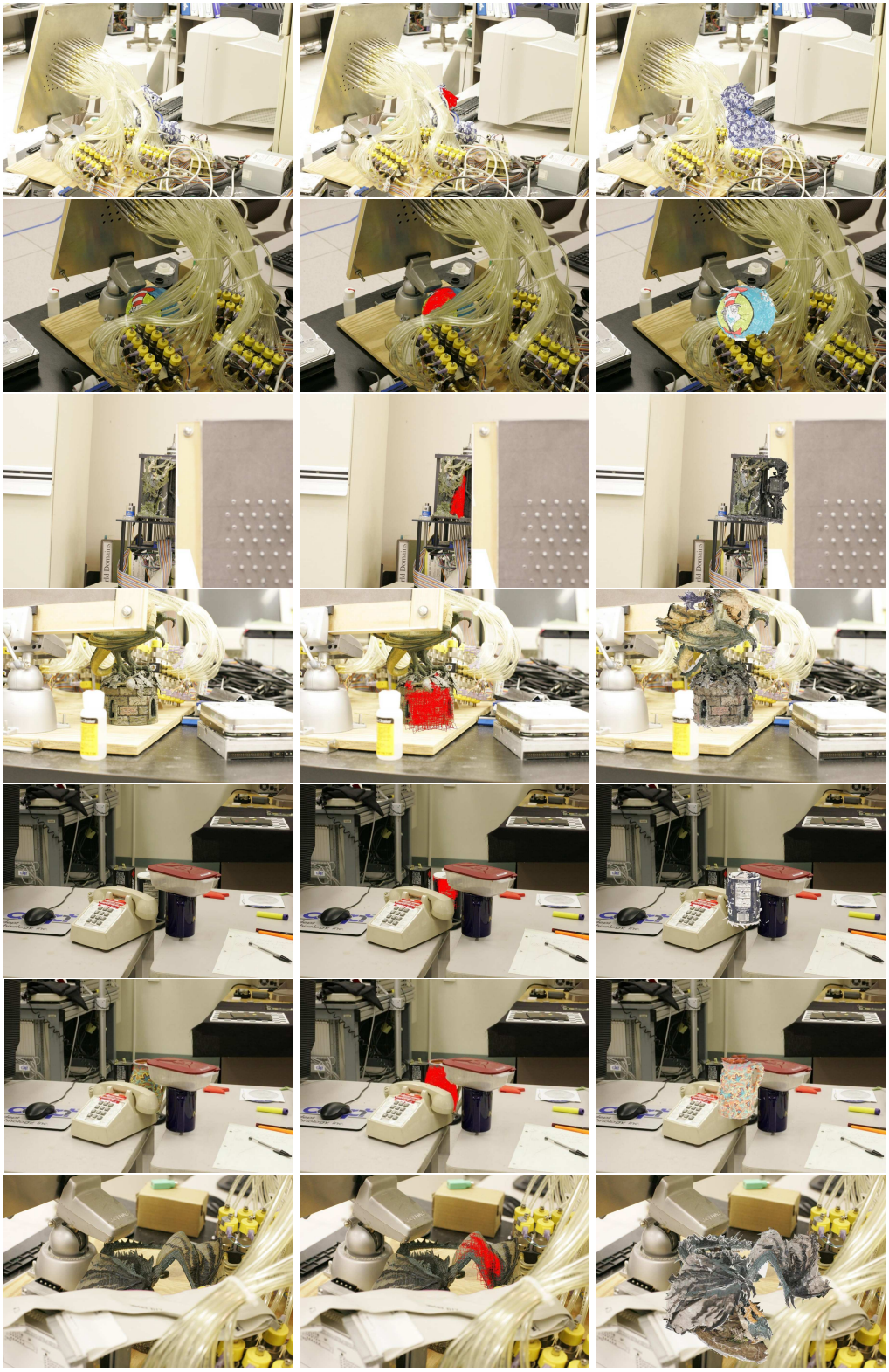
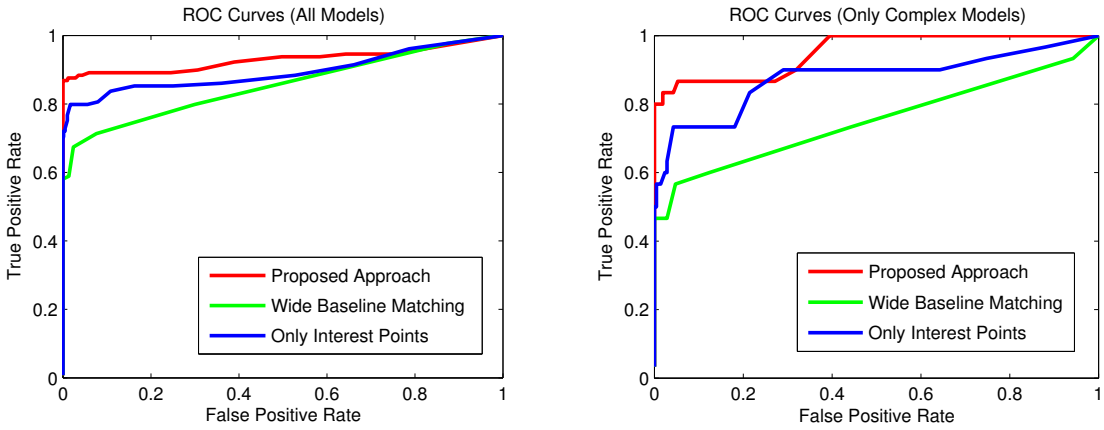


Figure 5.3: Left column: test image. Center column: matched patches. Right column: predicted location.



(a) ROC (all models).

(b) ROC (only Chest Buster, Dragon, Small Dragon).

Figure 5.4: Comparison ROC plots.

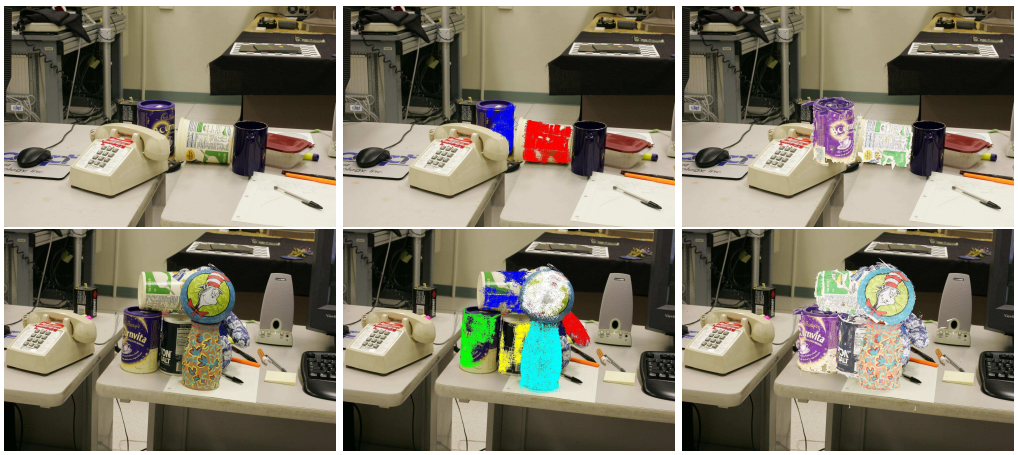


Figure 5.5: Left column: test image. Center column: matched patches. Right column: predicted location.

References

[1] J. L. Crowley and A. C. Parker. 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, 1984.

[2] V. Ferrari, T. Tuytelaars, and L. Van Gool. Integrating multiple model views for object recognition. In *Conference on Computer Vision and Pattern Recognition*, 2004.

[3] V. Ferrari, T. Tuytelaars, and L. Van Gool. Simultaneous object recognition and segmentation by image exploration. In *European Conference on Computer Vision*, 2004.

[4] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Communications ACM*, 24(6):381–395, June 1981.

[5] David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.

[6] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004. In press.

[7] David G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, pages 1150–1157, Corfu, Greece, 1999.

[8] David G. Lowe. Local feature view clustering for 3d object recognition. In *Conference on Computer Vision and Pattern Recognition*, 2001.

[9] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference*, volume I, pages 384–393, 2002.

[10] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, volume I, pages 128–142, 2002.

[11] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. submitted to IJCV.

[12] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. In *Conference on Computer Vision and Pattern Recognition*, 2003.

[13] Francis Schmitt Raouf Benjemaa. A solution for the registration of multiple 3d point sets using unit quaternions. In *European Conference on Computer Vision*, 1998.

[14] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3d object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In *Conference on Computer Vision and Pattern Recognition*, volume II, pages 272–277, 2003.

[15] Fredrick Rothganger. *3D object modeling and recognition in photographs and video*. PhD thesis, University of Illinois, Urbana Champaign, 2004.

[16] Tinne Tuytelaars and Luc J. Van Gool. Content-based image retrieval based on local affinely invariant regions. In *Visual Information and Information Systems*, pages 493–500, 1999.

[17] H. Voorhees and T. Poggio. Detecting textons and texture boundaries in natural images. In *International Conference on Computer Vision*, pages 250–258, 87.