Fusion of 3D and Appearance Models for Fast Object Detection and Pose Estimation

Hesam Najafi[1], Yakup Genc[1], and Nassir Navab[2]
{Hesam.Najafi|Yakup.Genc}@siemens.com, navab@cs.tum.edu

[1] Real-time Vision and Modeling Department,
Siemens Corporate Research, Inc.,
Princeton, NJ 08540, USA
[2] Institut für Informatik
Technische Universität München,
Boltzmannstr. 3, 85748 Garching bei München, Germany

Abstract. Real-time estimation of a camera's pose relative to an object is still an open problem. The difficulty stems from the need for fast and robust detection of known objects in the scene given their 3D models, or a set of 2D images or both. This paper proposes a method that conducts a statistical analysis of the appearance of model patches from all possible viewpoints in the scene and incorporates the 3D geometry during both matching and the pose estimation processes. Thereby the appearance information from the 3D model and real images are combined with synthesized images in order to learn the variations in the multiple view feature descriptors using PCA. Furthermore, by analyzing the computed visibility distribution of each patch from different viewpoints, a reliability measure for each patch is estimated. This reliability measure is used to further constrain the classification problem. This results in a more scalable representation reducing the effect of the complexity of the 3D model on the run-time matching performance. Moreover, as required in many real-time applications this approach can yield a reliability measure for the estimated pose. Experimental results show how the pose of complex objects can be estimated efficiently from a single test image.

1 Introduction

Estimating the pose of a camera relative to an object is one of the most studied problems in computer vision and photogrammetry. While reliable solutions have been proposed for pose estimation given correspondences [1–4] and feature-based 3D tracking [5–7], fully automated estimation of the initial camera's pose for tracking is still an open problem. The difficulty stems from the need for fast and robust detection of known objects in the scene given their 3D models, or a set of 2D images or both. Fast and robust pose estimation has a wide variety of applications, such as robot navigation, surveillance, and augmented reality.

Computer vision literature includes many object detection approaches [8, 9, 5, 10, 11] based on representing objects of interests by a set of local features which

are characterized by invariant descriptors for matching [12–16]. Combination of such descriptors provide robustness against partial occlusion and cluttered backgrounds. The descriptors are ideally invariant to viewpoint and illumination variations. Most of these methods make use of techniques for wide-baseline stereo matching solely based on 2D images without considering any run-time requirements. However, in many applications where real-time object detection is required both 3D models and several training images may be available or can be created easily during an off-line process.

This paper presents an alternative approach for fast object detection and pose estimation by fusing both 3D and appearance models. It shows that real-time performance can be achieved by using the underlying 3D information to limit the number of hypothesis for the robust matching process. Especially for large environments this renders our method very powerful. Our method differs in two aspects from the state of the art. First, we propose a statistical analysis and evaluation of the appearance and shape of features from all possible viewpoints in the scene combining real and synthetic viewpoints. Second, we make use of the known 3D geometry in both matching and pose estimation processes. We show that by fusing both appearance and geometric information rather than using them in separate procedures we can improve both time and functional performance, and make our approach more scalable for large environments.

Our approach has two phases. In the training phase, a compact appearance and geometric representation of the target object is built. This is as an off-line process. The second phase is an on-line process where a test image is processed for detecting the target object using the representation built in the training phase. During training, the variations in the descriptors of each feature are learned using principal component analysis (PCA). Furthermore, for each feature a reliability measure is estimated by analyzing the computed visibility distribution from different viewpoints. The problem of finding matches between sets of features in the test image and on the object model is then formulated as a classification problem which is constrained by using the reliability measure of each feature.

As an application, our method is intended to be used to provide robust initialization for a frame rate feature-based pose estimator [6] where robustness and time efficiency are very critical. In this case the initial pose recovery is sufficient to be performed under one second.

2 Previous work

A number of approaches have been proposed addressing the problem of 3D object detection for pose estimation. Some methods use statistical classification techniques, e.g PCA to compare the test image with a set of calibrated training images [17]. Others are based on matching of local image features [12, 13, 18, 19, 5, 20–24]. While some approaches use simple 2D features such as corners or edges, more sophisticated approaches rely on local feature descriptors which are insensitive to viewpoint and illumination changes. Usually geometric constraints are used as verification criteria of the estimated pose. Rothganger et al. [20] introduced a 3D object modeling and recognition algorithm for affine viewing conditions. Photometrically and geometrically consistent matches are selected in a RANSAC-based

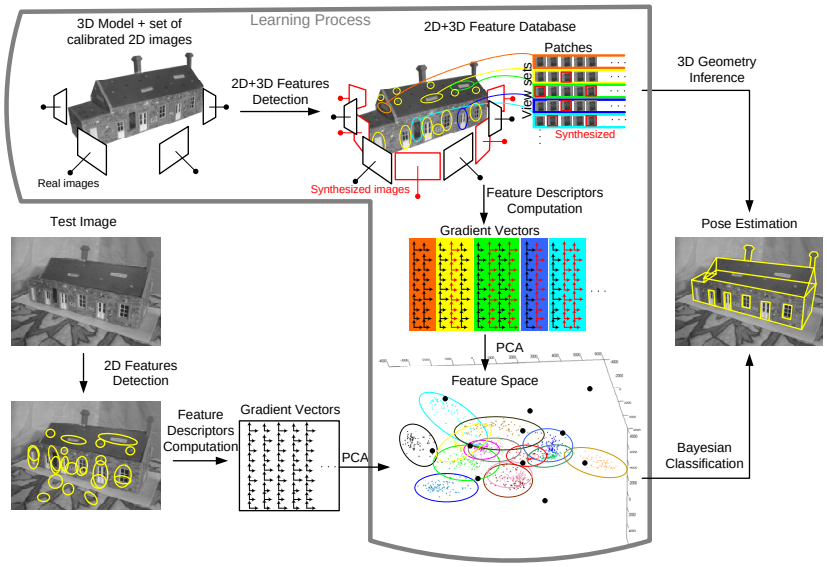

Fig. 1. Overview of the proposed object detection process for real-time pose estimation.

pose estimation procedure. Even though this method achieves good results for 3D object detection, it is too slow for real-time applications. Lepetit et al. [19] treat wide baseline matching of key points as a classification problem, where each class corresponds to the set of all possible views of each point. Once potential matches have been established they apply a plain RANSAC method to recover the 3D pose. Recently they introduced an approach for object pose estimation in real-time [25], where randomized trees are used as the classification technique. Keypoint recognition relies solely on 2D image intensity values within small windows around these keypoints.

3 Proposed approach

Our goal is to automatically detect objects and recover their pose for arbitrary images (*test image*). The proposed object detection approach is based on two stages: A learning stage which is done off-line and the matching stage at run-time. The entire learning and matching processes are fully automated and unsupervised. Sections 3.1 and 3.2 describe the learning step in more detail. In Sections 3.3 and 3.4 we introduce the matching and pose estimation algorithms that enforce both photometric and geometric consistency constraints.

3.1 Creating view sets based on similarity maps

In the first step of the learning stage a set of stable feature regions are selected from the object by analyzing their detection repeatability and accuracy as well as their visibility from different viewpoints.

Images represent a subset of the sampling of the so called *plenoptic function* [26]. The plenoptic function is aparameterized function for describing everything that can be seen from all possible viewpoints in the scene. In computer graphics

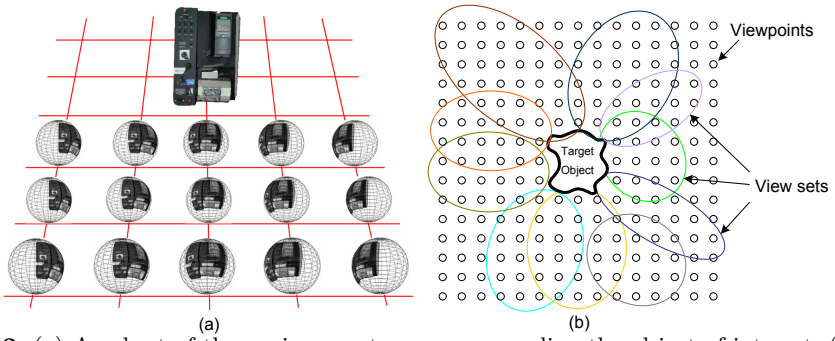

Fig. 2. (a) A subset of the environment maps surrounding the object of interest. (b) A 2D illustration of the 3D clusters of the view sets surrounding the target object.

terminology the plenoptic function describes the set of all possible *environment maps* for a given scene. In our case, we define a complete sample of the plenoptic function as a full spherical environment map (see Fig. 2(a)). Having a set of calibrated images and the virtual model of the target object, the viewing space is coarsely sampled at discrete viewpoints and a set of environment maps is created. Since not all samplings can be covered by the limited number of training images, synthesized views are created from other viewpoints using computer graphics rendering techniques.[3] Next, affine covariant features [27] are extracted from the environment maps. In our experiments we use a variant of Hessian- and Harris-affine detector introduced in [15]. We also tested the scale and rotation invariant SIFT detector [10] (see section 4). We then select "good" feature regions which are characterized by their detection repeatability and accuracy. The basic measure of accuracy and repeatability is based on the relative amount of overlap between the detected regions in the environment maps and the respective reference regions projected onto that environment map using the ground truth transformation. The reference regions can be determined e.g. from the parallel views to the corresponding feature region on the object model (*model region*). This *overlap error* is defined as the error in the image area covered by the respective regions [15].

For each model region a *view set* is the set of its appearances in the environment maps from all possible viewpoints (see Fig. 2(b)). Depending on the 3D structure of the target object a model region may be clearly visible only from certain viewpoints in the scene. We create for each model feature a *similarity map* by comparing it with the corresponding extracted features. As a similarity measure we use the Mahalanobis distance between the respective SIFT descriptors. For each model region the respective similarity map represents its visibility distribution. This analysis can also be used to remove the repetitive features visible from the same viewpoints in order to keep the more distinctive features for matching. Based on the similarity maps of each model region we cluster groups of viewpoints together using the mean-shift algorithm [28]. The clustered viewpoints for a model region m_j are $W(m_j) = \{v_{j,k} \in \Re^3 | 0 < k \leq N_j\}$, where $v_{j,k}$ is a viewpoint of

[3] Due to complexity of the target object and the sampling rate this can be a time consuming procedure. However, this does not affect the computational cost of the system at run-time since this can be done off-line.

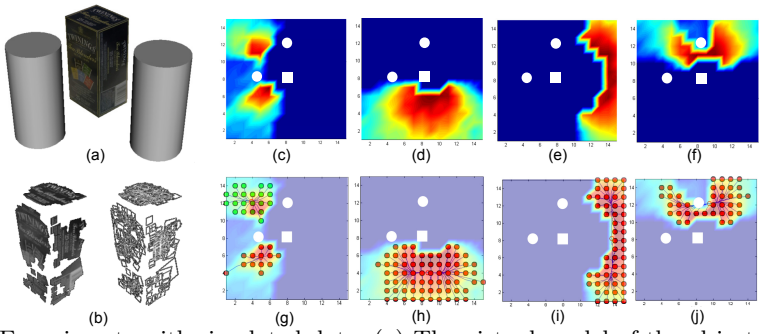

Fig. 3. Experiments with simulated data. (a) The virtual model of the object. (b) The extracted features on the model. (c)-(f) Top-down view of a subset of the similarity maps. (g)-(j) The clustered view sets using mean-shift algorithm.

that region. Figure 3 shows some results of a simulated scene including a box and two cylinders. The faces of the box are rendered with the texture obtained from a real tea box. Figure 3(c)-(f) show top down views of a subset of the similarity maps of four patches selected from each side of the box. Note how the presence of an occluding object (cylinders) is reflected in the similarity maps. The respective view sets determined by mean shift clustering are shown in Fig. 3(g)-(j).

3.2 Learning the statistical representation

This section describes a method to incorporate multiple view descriptors of each view set into our statistical model. We use the PCA-SIFT descriptor [29] for a more compact representation (e.g. first 32 components). To minimize the impact of variations of illumination, especially between the real and synthesized images, the descriptor vectors are normalized to unit magnitude. The image gradient vectors $g_{i,j}$ are projected into the feature space to a feature vector $e_{i,j}$.

We suppose that the distribution of the gradient vectors is Gaussian for the carefully selected features as described in the previous section. For each region we take k samples from the respective environment maps so that the distribution of their feature vectors $e_{i,j}$ for $0 < j \leq K$ in the feature space is Gaussian. To ensure the Gaussian distribution of the gradient vectors for each view set we apply the χ^2 test for a maximal number of samples. If the χ^2 test fails after a certain number of samplings for a region, the region will be considered as not reliable enough and will be excluded. For each input view set V_i we then learn the covariance matrix Σ_i and the mean μ_i of the distribution.

3.3 Matching as a classification problem

Matching is the task to find groups of corresponding pairs between the regions extracted from the model and test image, that are consistent with both appearance and geometric constraints. The matching problem can be formulated as a classification problem [19]. Our goal is to construct a classifier so that the misclassification rate is low. From the test image, the features are extracted in the same manner as in the learning stage and their gradient image vectors are computed. The descriptors are then projected into feature space using PCA (bold dots in Fig.

1). We use the Bayesian classifier to decide whether a test descriptor belongs to a view set class or not. Let $C = \{C_1, ..., C_N\}$ be the set of all classes representing the view sets and let F denote the set of 2D-features $F = \{f_1, ..., f_K\}$ extracted from the test image. Using the Bayesian rule the *a posteriori probability* $P(C_i|f_j)$ for a test feature f_j that it belongs to the class C_i is calculated as

$$P(C_i|f_j) = \frac{p(f_j|C_i)P(C_i)}{\sum_{k=1}^{N} p(f_j|C_k)P(C_k)}. \quad (1)$$

We compute for each test descriptor the a posteriori probability of all classes and select candidate matches using thresholding. Let $m(f_j)$ be the respective set of most probable potential matches $m(f_j) = \{C_i|P(C_i|f_j) \geq T\}$. The purpose of this threshold is only to accelerate the run-time matching and not to consider matching candidates with low probability. However this threshold is not crucial for the results of pose estimation.

3.4 Pose estimation using geometric inference

This section describes a method using geometric consistency to constrain the search space for finding candidate matches. For the pose estimation a set of $N \geq 3$ matches are required. In an iterative manner we choose the first match $f'_1 \leftrightarrow C'_1$ as the pair of correspondences with the highest confidence:

$$\underset{\substack{f_k \in F \\ C_l \in C}}{argmax} P(C_l|f_k).$$

We define V_{C_l} as the set of all classes of regions which should also be visible from the viewpoints where C_l is visible

$$V_{C_l} = \{C_k \in C||W_k \cap W_l| \neq 0\},$$

where W_j is the set of 3D-coordinates of the clustered viewpoints $\{v_{j,k}|0 < k \leq N_j\}$ for which the respective model region is visible (see building environment maps, Section 3.1).

Assuming the first candidate match is correct, the second match $f'_2 \leftrightarrow C'_2$ is chosen only from the respective set of visible regions. Therefore after each match selection the search area is constrained to visibility of those regions based on previous patches. In general the k^{th} candidate match $f'_k \leftrightarrow C'_k, 1 < k \leq N$ is selected in a deterministic manner

$$(f'_k, C'_k) = \underset{\substack{f_k \in F \setminus \{f_1, ..., f_{k-1}\} \\ C_k \in \bigcap_{l=1}^{k-1} V_{C'_l}}}{argmax} P(C_k|f_k).$$

The termination criteria is defined based on the back-projected overlap error (see Section 3.1) in the test image. This algorithm can be implemented in different ways. One way is a recursive implementation with an interpretation tree where the nodes are visited in the depth-first manner. The depth is the number of required matches N for the pose estimation method. This algorithm has a lower complexity as the results will show, than the plain version of RANSAC or the "exhaustive" version where all pairs of candidate matches are examined.

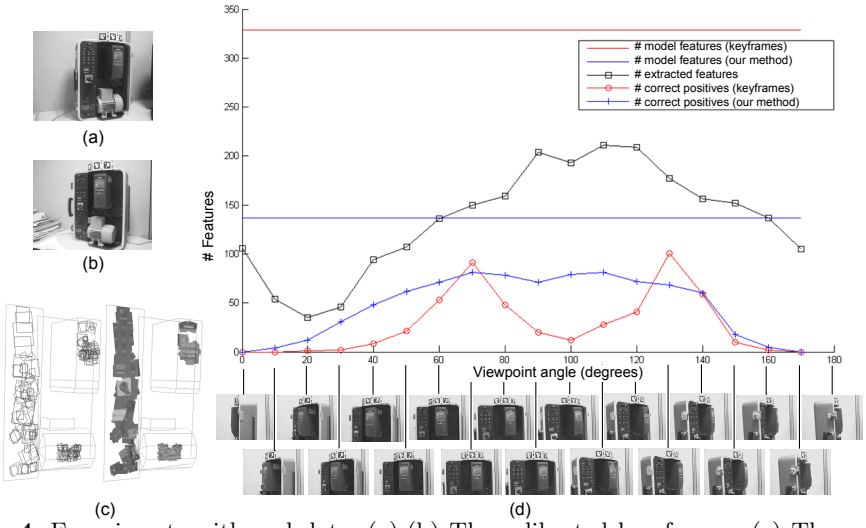

Fig. 4. Experiments with real data. (a)-(b) The calibrated key frames. (c) The set of most visible patches extracted on the model based on the statistical analysis using the similarity maps. (d) Metrics used to compare the results (see text).

4 Experimental results

The proposed method has been tested in a series of experiments using virtual and real objects. Due to the space limitations we only present a subset of the results using real objects. The off-line learning process uses ImageModeler from RealViz [30] to obtain a 3D model.[4] Our experimental setup consists of a target object and a commonly available FireWire camera (Fire-I). The camera is internally calibrated and lens distortions are corrected using the Tsai's algorithm [31].

We conducted a set of experiments to analyze the functional and the timing performance of our approach. The results were compared against a conventional approach based solely on 2D key frames. Our approach requires an input consisting of a set of images (or key frames) of the target object. One target object is shown in Fig. 4(a). The key frames were calibrated. We used a calibration object (a known set of markers) for automatically calibrating the views. These markers

[4] The accuracy requirements depend on the underlying pose estimation algorithms, the object size and the imaging device.

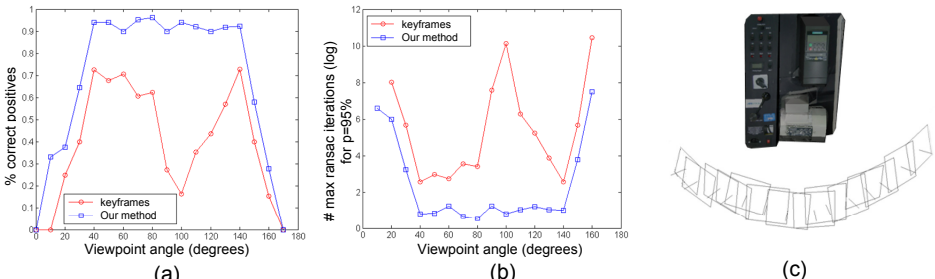

Fig. 5. Experiments with real data. (a)-(b) Performance evaluation (see text). (c) Visualization of the pose estimation results.

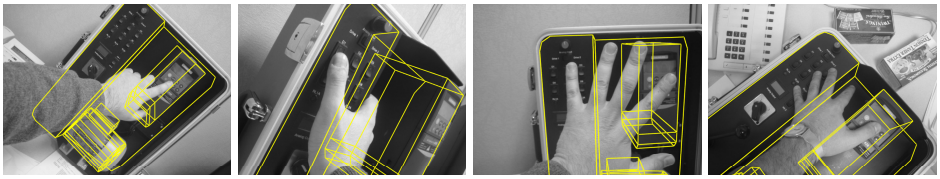

Fig. 6. Experiment 1: Control Box. Pose estimation results on test images.

were used to compute the ground truth for evaluating the matching results on test frames as well.

In the first experiment, we analyzed the functional performance against view point variations for the same scene but under uncontrolled lighting. The images were taken by a moving camera around the object. For the sake of clarity of presentation, we show a subset of 19 test images from this sequence with additional two images as key frames (see Fig. 4(a)-(b),(d)). All those images were calibrated as explained above. Fig. 4(d) shows some metrics we used to compare these results. One measure of performance is the final size of the representation (number of features in the database) used for both methods indicated by the two straight lines. With increasing number of key frames the size of the database in the conventional case would increase linearly with the number of key frames. In contrast, our method keeps fewer features in the 2D-3D database after careful implicit analysis of their planarity, visibility and detection repeatability. The database size in our method is proportional to the scene complexity not the number of available key frames. This is an important property for the scalability of the system for more complex objects. Fig. 4(d) also shows the number of extracted features and the number of correct matches found by both methods for each of the 19 test images. It should be noted that, near the two key frames our method obtains less correct matches compared to the conventional method. This is due to the fact that our representation generalizes the extracted features whereas the conventional methods keeps them as they are. The generalization has the cost of missing some of the features in the images closer to the key frames. On the other hand, the generalization helps to correctly match more features in disparate test views.

Complexity and performance of robust pose estimation methods like RANSAC are dependent not on the number of correct matches but the ratio between correct and false matches. Fig. 5(a) shows the percentage of correct matches vs the viewing angle for the proposed method and the conventional approach. Although near the key frames our method obtains fewer matches, it has a higher percentage of correct positives. As a result of this and the visibility constraints used our method needs only a few RANSAC iterations for pose estimation. This brings

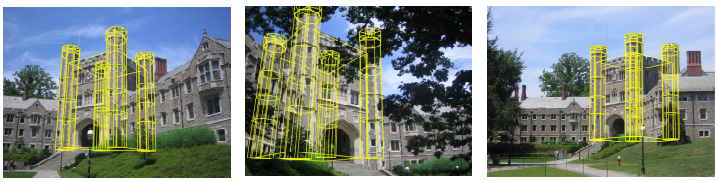

Fig. 7. Experiment 2: Blair Tower. Pose estimation results on test images.

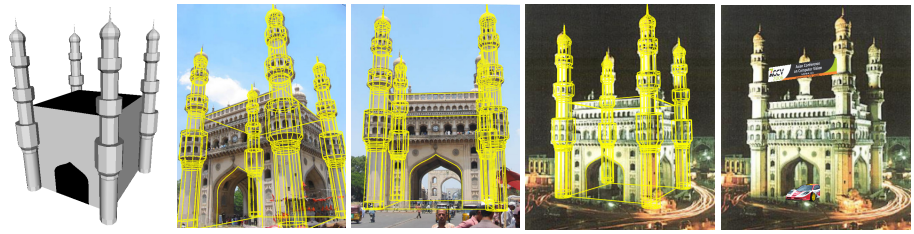

Fig. 8. Experiment 3: Char Minar. (a) 3D model. (b)-(d) Pose estimation results on test images, and with virtual objects (e).

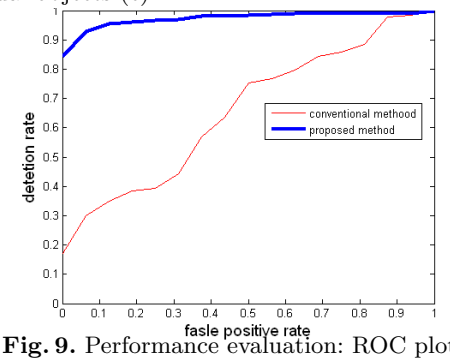

Fig. 9. Performance evaluation: ROC plot.

us to the timing performance of the matching methods. We use a more complex matching method than the conventional one. Therefore, each individual match costs more. However, with increasing complexity of the target object with respect to self-occlusions our representation becomes more efficient. Fig. 5(b) shows the respective maximal number of iterations needed (logarithmic scale) for RANSAC based pose estimation with a confidence probability of 95%. Fig. 5(c) shows a visualization of the pose estimation results. We obtain up to five folds speed-up compared to the exhaustive RANSAC method. Our non-optimized implementation needs about 0.3 to 0.6 second compared to 2.5 seconds for the conventional approach. In Fig. 6 (a)-(d) more results are shown for experiments using test images with occlusions, cluttered background and illumination changes. The detection results are quite robust and the estimated pose is accurate enough to initialize our real-time 3D tracker [6]. Fig. 8 and 7 show the results of two other experiments in outdoor environments. We used each time two images to build a coarse 3D model and applied our method to several test images.

The performance of the matching part of our system was evaluated by processing all pairs of object model and test images, and counting the number of established matches. Fig. 9 shows the ROC curve that depicts the detection rate vs false-positive rate, while varying the detection threshold T. Compared to the keyframe-based approach the proposed approach performs very well and achieves 97% detection with 5% false-positives.

5 Conclusions

This paper addressed the problem of real-time object detection for pose estimation. The major contribution of this paper is the integration of the known 3D

geometry of the target model during both matching and pose estimation steps. This is achieved by a statistical analysis of the appearances distribution of model patches in the viewing space. Instead of the local planarity assumption used in previous approaches, our proposed method is able to learn the visibility distribution of the variations in the local descriptors considering their known geometry.

Acknowledgments. We would like to thank D. Lowe for providing his implementation of the SIFT Keypoint Detector. We also would like to thank G. Klinker, Y. Tsin and V. Ramesh for helpful discussions during the course of this work.

References

1. Dementhon, D., Davis, L.S.: Model-based object pose in 25 lines of code. ECCV (1992)
2. Pollefeys, M., Koch, R., Van Gool, L.: Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. ICCV (1998)
3. Nister, D.: An efficient solution to the five-point relative pose problem. CVPR (2003)
4. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2004)
5. Vacchetti, L., Lepetit, V., Fua, P.: Stable real-time 3d tracking using online and offline information. PAMI (2004)
6. Genc, Y., Riedel, S., Souvannavong, F., Akinlar, C., Navab, N.: Marker-less tracking for ar: A learning-based approach. ISMAR (2002)
7. Davison, A., Murray, D.: Simultaneous localization and map-building using active vision for a robot. PAMI (2002)
8. Ferrari, V., Tuytelaars, T., Van Gool, L.: Integrating multiple model views for object recognition. CVPR (2004)
9. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: Segmenting, modeling, and matching video clips containing multiple moving objects. CVPR (2004)
10. Lowe, D.: Distinctive image features from scale-invariant key points. IJCV (2004)
11. Meltzer, J., Soatto, S., Yang, M.H., Gupta, R.: Multiple view feature descriptors from image sequences via kernel principal component analysis. ECCV (2004)
12. Schmid, C., Mohr, R.: Local gray value invariants for image retrieval. PAMI (1997)
13. Lowe, D.G.: Object recognition from local scale-invariant features. ICCV (1999)
14. Van Gool, L., Moons, T., Ungureanu, D.: Affine/photometric invariants for planar intensity patters. ECCV (1996)
15. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. ECCV (2002)
16. Schaffalitzky, F., Zisserman, A.: Multi-view matching for unordered image sets, or how do i organize my holiday snaps?. ECCV (2002)
17. Nayar, S.K., Nene, S.A., Murase, H.: Real-time 100 object recognition system. PAMI (1996)
18. Li, Y., Tsin, Y., Genc, Y., Kanade, T.: Object detection using 2d spatial ordering constraints. CVPR (2005)
19. Lepetit, V., Pilet, J., Fua, P.: Point matching as a classification problem for fast and robust object pose estimation. CVPR (2004)
20. Rothganger, F., Lazebnik, S., Schmid, C., Ponce, J.: 3d object modeling and recognition using affine-invariant patches and multi view spatial constraints. CVPR (2003)
21. Tuytelaars, T., VanGool, L.: Wide baseline stereo matching based on local, affinely invariant regions. BMVC (2000)
22. Allezard, N., Dhome, M., Jurie, F.: Recognition of 3d textured objects by mixing view-based and model-based representations. ICPR (2000)
23. Jurie, F.: Solution of the simultaneous pose and correspondence problem using gaussian error model. CVIU (1999)
24. Mindru, F., Moons, T., VanGool, L.: Recognizing color patterns irrespective of viewpoint and illumination. CVPR (1999)
25. Lepetit, V., Lager, P., Fua, P.: Randomized trees for real-time keypoint recognition. CVPR (2005)
26. Adelson, E.H., Bergen, J.R.: Computational models of visual processing, chapter 1: The plenoptic function and the elements of early vision. The MIT Press, Cambridge (1991)
27. Mikolajczyk, K., Tuytelaars, T., Schmid, C. Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. IJCV (2004)
28. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI (2002)
29. Ke, Y., Sukthankar, R.: Pca-sift: A more distinctive representation for local image descriptors. CVPR (2004)
30. RealViz. (www.realviz.com)
31. Tsai, R.Y.: A versatile camera calibration technique for high-accuracy 3d machine vision metrology using of the shelf tv cameras. IEEE Journal of Robotics and Automation (1987)