Proceedings of the Ninth IAPR Conference on Machine Vision Applications, Tsukuba, Japan, 2005, pp. 5-8

3-D Object Tracking with the Adaptive Hyperplane Approach Using SIFT Models for Initialization

Christoph Gräßl, Timo Zinßer, Ingo Scholz, and Heinrich Niemann

Universität Erlangen–Nürnberg
Lehrstuhl für Mustererkennung
Martensstraße 3, 91058 Erlangen
Germany

email: graessl@informatik.uni-erlangen.de

Abstract

Object tracking is still a challenging task, especially if it is done in a realistic environment. The ongoing increase of computational power and the efficiency of the algorithms allow real-time estimation of the object's pose in six degrees of freedom. One of these algorithms is the 3-D hyperplane approach, which is used throughout this paper, as it has been proven to be fast and accurate. We show how to enhance its robustness by using a linear illumination model to gain more insensitivity to variations of the illumination conditions. We also present an adaption to compensate appearence changes in case of external rotations.

Although some "six degrees of freedom" trackers have been established, the necessary initialization is often ignored or is only solved rudimentarily. In contrast to this, we show how to use a 3-D SIFT object model for initialization of the whole tracking system and prove its efficiency by experimental results using real image sequences.

1 Introduction

Visual object tracking has emerged as an important component in computer vision fields such as intelligent human machine interaction, surveillance, video annotation, and medical applications. The main purpose of tracking systems is the estimation of the position of an object in each image of an image sequence, under the assumption that the movements are small. Depending on the system demands, different solutions have been developed in the last decades, which allow the tracking of moving objects in a real environment with cluttered background using a non-fixed camera. Approaches based on color histograms [2, 11] have been proven to be very robust even in case of occlusions and strong appearance changes. However, they lack the ability to estimate object rotation. A different approach based on the eigenspace representation of an object [1] estimates the translation, rotation, and scale of an object in the image plane, but does not have the capability of real-time processing.

Template matching techniques that are based on a first order approximation of the object's motion [5, 9] are able to compute the translation, rotation, scale, and perspective distortion of an object in the image plane and are robust against appearance changes caused by illumination variations [4]. As many template matching approaches assume a planar surface, [8, 13] apply a 3-D model of an object and estimate the three translation and three rotation parameters of the object with known intrinsic camera parameters. Both approaches yield very good results, but for experimental evaulations only objects with primitive surfaces like planes and cylinders are used. In contrast,the approach of [15] allows tracking of arbitrary rigid objects by applying lightfield models in a probabilistic approach, but lacks in computational efficiency if all six pose parameters have to be calculated.

Our tracking system is based on the *3-D hyperplane approach* by [8], a 3-D template matching technique. We present the integration of 3-D point models which are acquired by a *structure-from-motion* approach [6] from arbitrary objects. The correspondence problem is solved by using *SIFT features* [10], which we also use for the initialization of the object tracker, since the initial pose is generally not known. Therefore, corresponding feature points of the initial image and the 3-D point model of the object are detected and the six pose parameters are estimated by the *POSIT* algorithm [3]. As the correspondences of feature points can be incorrect, we use the *LMedS* [12] to compensate for outliers. We examine the capabilities and limitations of this method by experiments with real images.

The appearance of an object can change rapidly due to illumination variations, e.g., caused by auto-exposure correction of the camera. We propose to compensate those influences with a linear illumination model. Another enhancement addresses the problem of the appearance change of an object caused by *external rotation* (i.e., rotation not in the image plane). We show how to adapt the model during runtime to incorporate new views to enhance the robustness of the motion estimation. In our experiments, we use an object with a complex surface and show that our proposed method yields very good results, even in scenes with cluttered background.

2 Template Matching with Hyperplanes

Template matching algorithms for data-driven tracking work on a sequence of images, where every image is indexed by a discrete time t. Additionally, a *reference template* must be specified in the first image. The reference template is defined by the vector $\boldsymbol{r} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)^T$, which contains the homogeneous 3-D coordinates [6] of selected object points. The gray-level intensity of a 3-D point $\boldsymbol{x}_i = (x_i, y_i, z_i, 1)^T$, which has been projected into the image plane at time t using the projection matrix of a calibrated camera, is given by $f(\boldsymbol{x}, t)$. Consequently, the vector $\boldsymbol{f}(\boldsymbol{r}, t)$ contains the intensities of template $\boldsymbol{r}$ at time t.

The transformation of the reference template $\boldsymbol{r}$ at time t is modeled by $\boldsymbol{r}(t) = \boldsymbol{g}(\boldsymbol{r}, \boldsymbol{\mu}(t))$, where the vector $\boldsymbol{\mu}(t) = (\mu_{t_1}(t), \mu_{t_2}(t), \mu_{t_3}(t), \mu_{r_1}(t), \mu_{r_2}(t), \mu_{r_3}(t))^T$ contains the 3-D translation parameters $\mu_{t_1}(t), \mu_{t_2}(t), \mu_{t_3}(t)$ and the three rotation parameters $\mu_{r_1}(t), \mu_{r_2}(t), \mu_{r_3}(t)$ (axis-angle parameterization [6]) of the object. Template matching can now be described as computing the motion parameters $\boldsymbol{\mu}(t)$ that minimize the least-squares intensity difference between the reference template and the current template.

Since non-linear minimization in a high-dimensional parameter space involves extremely high computational cost, it is more efficient to use a first order approximation

$$\begin{aligned}\boldsymbol{\mu}(t+1) &= \boldsymbol{\mu}(t) + \\ &\quad \boldsymbol{A}(t+1)\left(\boldsymbol{f}(\boldsymbol{r}, t_0) - \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{r}, \boldsymbol{\mu}(t)), t+1)\right)\end{aligned} \quad (1)$$

as presented in [5, 9]. The transformation function $\boldsymbol{g}(\boldsymbol{r}, \boldsymbol{\mu}(t))$, which projects the model points into the image plane, is given by

$$\boldsymbol{g}(\boldsymbol{r}, \boldsymbol{\mu}(t)) = \left(\boldsymbol{M}_i \left(\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{smallmatrix}\right) \boldsymbol{M}_e(\boldsymbol{\mu}(t)) \boldsymbol{r}^T\right)^T, \quad (2)$$

where the matrix $\boldsymbol{M}_i \in \mathbb{R}^{3\times3}$ contains the intrinsic camera parameters and $\boldsymbol{M}_e(\cdot) \in \mathbb{R}^{4\times4}$ contains the extrinsic camera parameters.

There are two approaches for computing the matrix $\boldsymbol{A}(t)$ from Eq. (1). Hager and Belhumeur [5] propose the application of a Taylor approximation. The hyperplane approach presented in [9] acquires matrix $\boldsymbol{A}(t)$ by a least-squares estimation which is done in a short initialization step. It was also shown how to make matrix $\boldsymbol{A}$ independent of time t. As the hyperplane approach has a superior basin of convergence, we will use it throughout the rest of this paper.

3 SIFT Object Models

The acquisition of the reference template $\boldsymbol{r}$ is a very challenging task. We decided to use a structure-from-motion technique [6], because many robust algorithms are known and only a short sequence of *training* images is required to create a precise point model of the object. For solving the correspondence problem of 2-D points, we use local SIFT features [10], which consist of a 2-D coordinate (feature point) and a 128 dimensional feature vector $\boldsymbol{c}$. The SIFT feature points are detected by applying a scale selection mechanism based on differences of Gaussian smoothed images. For detailed information refer to the original paper [10].

For every feature point in every training image, a SIFT feature vector $\boldsymbol{c}$ is calculated. In order to estimate the 3-D position of the feature points, similar features are collected in a set

$$\begin{aligned}\boldsymbol{C}_i &= \{\boldsymbol{c} \mid m(\boldsymbol{c}_k) \neq m(\boldsymbol{c}_l) \wedge \exists d(\boldsymbol{c}_k, \boldsymbol{c}_l) < \epsilon\}; \quad (3)\\ & \quad i \neq j \Rightarrow \boldsymbol{C}_i \cap \boldsymbol{C}_j = \emptyset,\end{aligned}$$

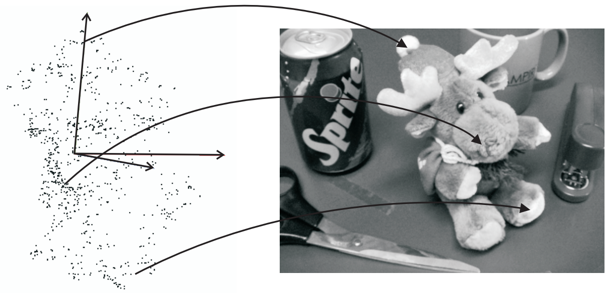


Figure 1: Example of the assignment of the 3-D point model (left) of a SIFT object model to a 2-D image. The pose is estimated by the POSIT algorithm [3].

where $m(\boldsymbol{c})$ returns the index of the image where the feature vector $\boldsymbol{c}$ has been calculated, i, j are indices of the set, $d(\boldsymbol{c}_k, \boldsymbol{c}_l)$ is the Euclidean distance of two features and ϵ is a threshold in order to ensure that only similar features are stored in the set. This set is very similar to the so called *trail* in [7]. For every set $\boldsymbol{C}_i$, we estimate the 3-D position $\boldsymbol{x}_i$ by the structure from motion algorithm of [7, Section 3] and calculate a mean feature vector $\bar{\boldsymbol{c}}_i$. The reference template $\boldsymbol{r}$ is built using all of these 3-D points.

In contrast to point tracking methods, which are very commonly used for 3-D reconstruction, the application of the SIFT features has the advantage that it can be used for both estimation of the reference template and initialization of the object tracker. In principle, the initialization of the tracker is similar to the calibration problem, as the 3-D point model represents the calibration pattern and for every point $\boldsymbol{x}_i$ a mean SIFT feature $\bar{\boldsymbol{c}}_i$ has been calculated. After the extraction of the SIFT features of the reference image, the assignment of a feature vector $\boldsymbol{c}$ to the n-th model feature vector is done by

$$n(\boldsymbol{c}) = \operatorname*{argmin}_i d(\boldsymbol{c}, \bar{\boldsymbol{c}}_i) \ . \quad (4)$$

In addition, assignments for which the Euclidean distance exceeds the threshold ϵ of Eq. (3) are ruled out. We estimate the initial object position $\boldsymbol{\mu}(t_0)$ for the image at time $t = 0$ using the POSIT algorithm [3], but in principle, more complex techniques are applicable as well. This initialization step is illustrated in Fig. 1.

4 Improving the Robustness

The change of appearance is an important challenge in template matching approaches. One reason for those changes are illumination variations. We apply a normalization of the template's intensity distribution using its mean and variance. This approach has been proven to be very efficient with regard to robustness and computating time [4].

A second reason for appearance changes, and thus low robustness of the tracker, are large external rotations. We enhance the tracking system by training separate approximation matrices $\boldsymbol{A}$ for different views during runtime. Consequently, the approximation matrix which has been calculated at the viewpoint with the external rotation most similar to $\boldsymbol{\mu}(t-1)$ is used for estimating $\boldsymbol{\mu}(t)$.

As described in the previous section, the initial pose parameters $\boldsymbol{\mu}(t_0)$ are estimated by the POSIT algorithm. For

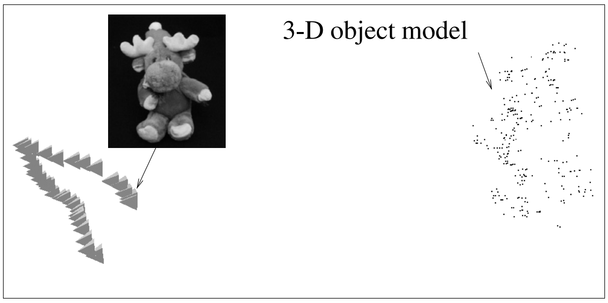


Figure 2: A video sequence of 36 images is used for a 3-d reconstruction of a toy-elk. The estimated 3-D object model (dots) and the camera positions (pyramids) of the 36 images are illustrated.

this, 2-D image points and their corresponding 3-D model points are required. Although local SIFT features are well suited for solving the correspondence problem, wrong assignments may occur and have to be taken into account. To reject these outliers, the LMedS algorithm [12] is applied.

5 Experiments

Many experiments with real image sequences have been performed to demonstrate that our proposed method leads to highly accurate tracking results. For this paper, we captured a video sequence (35 images) of a toy elk from different views for model acquisition with a hand-held Sony DFW-VL500 camera (resolution of 640 x 480 pixels). One property of this object is that the surface is highly complex and simple geometric models as in [8, 13] are ineligible. The toy elk was placed on a black cloth to prevent the extraction of feature points on the background. The SIFT feature point detector acquired 5495 feature points for the 36 images, consequently the average was 152.6 feature points per image (minimum 125, maximum 176). After detection of corresponding feature points (cf. Eq. (3)) and 3-D reconstruction, a model consisting of 306 3-D points was created. The computation time for calculating the SIFT features, detection of correspondences and 3-D reconstruction was 49 seconds on a 2.4 GHz Intel Pentium 4 PC. The result is presented in Fig. 2 where the 3-D model, camera positions of the corresponding 36 images, and one image of the image sequence are shown.

For demonstrating the capability of the tracker, we removed the black cloth and put some other objects into the scene to prove that our approach is not affected by a cluttered background. The estimation of $\boldsymbol{\mu}(t_0)$ and initialization of the hyperplane tracker took about 3 seconds. This value depends strongly on the number of detected feature points, and the computation time decreases in case of simple scenes. Typically, the computation time is between 1 and 4 seconds. After initialization, the estimation of $\boldsymbol{\mu}(t)$ (cf. Eq. (1)) is very efficient, and allows processing of 30 frames per seconds. Additionally, we moved the camera as well as the toy elk to different positions. Even in this case, the back-projected point model (using Eq. (2)) remains on the object. Some images of the whole sequence are presented in Fig. 3. We tested the presented approach successfully on other objects like cups, tetrapacks, tins, and books. Without an adaption step (c.f. Sec. 4), an external rotation of about 10 to 20 degrees is accepted by the tracker. The approach of [15] is not affected by external rotation, but lacks in real-time capability if all pose parameters have to be calculated. In contrast to that, the hyperplane approach is significantly more efficient in computing time and allows a fast initialization using the SIFT object model.

As the initialization plays an important role in our framework, we tested the efficiency of the pose estimation using the SIFT object model. For this we acquired a model of a package of juice and captured three image sequences each with 100 images with homogeneous, slightly cluttered, and highly cluttered background (Fig. 4). The pose was estimated twice for every image as proposed in Sec. 3, the first time using the LMedS and the second time not using it. The median backprojection error of the matched model feature points for every pose estimation, which is in our point of view a good quality measurement, has been calculated. For easier comparison, all median backprojection errors for one sequence have been ordered ascendingly. The average number of matched features in case of homogeneous background is about 69 and in case of cluttered background about 53. Even in the case of highly cluttered background, it can be seen in Fig. 4 (d) that the backprojection error is less than 1.5 pixels in about 80 percent of the estimations. The second graph (e) shows the benefits of the LMedS algorithm. It is clearly visible that this method enhances the detection accuracy.

6 Conclusion

In this paper, we presented a model-based tracking algorithm for estimating the object's 3-D pose. This technique is based on Jurie's hyperplane approach [8]. We addressed two problems, which arise using this method. The first problem is model acquisition and the second one is the initialization of the tracker. Both issues are solved by using local SIFT features [10].

A disadvantage of this approach is that model points could leave the field of view because of strong external rotation. This issue can be solved by using separate approximation matrices and regions which are related to visible points. Another idea would be to track each point individually with a point tracker like the Shi-Tomasi-Kanade tracker. Hidden points have to be rejected in this approach as well. For detection of these points, we plan to calculate a triangle net for the object model and verify the visibility by ray-tracing. The algorithm of [14] could be good starting point.

Acknowledgements

This work was partially funded by the European Commission 5th IST Programme - Project VAMPIRE and the German Science Foundation (DFG) under grant SFB 603/TP C2. Only the authors are responsible for the content.

We would like to thank David Lowe for providing the source code of the SIFT feature detector [10].

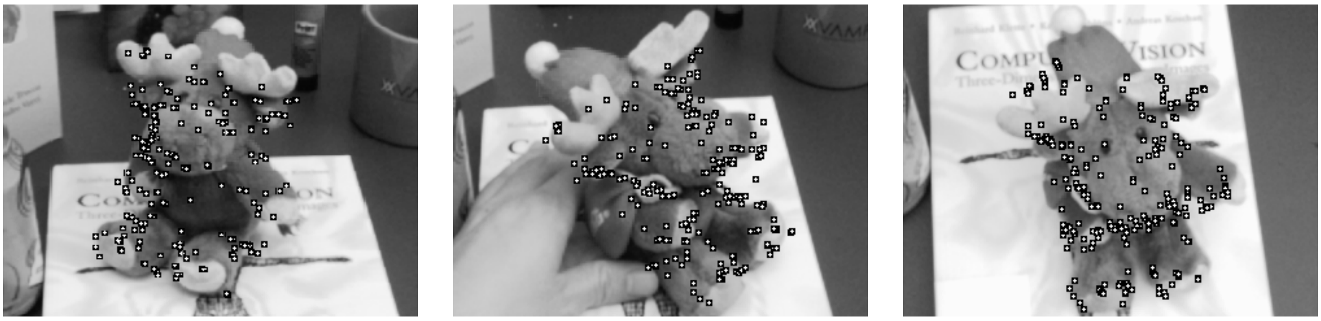
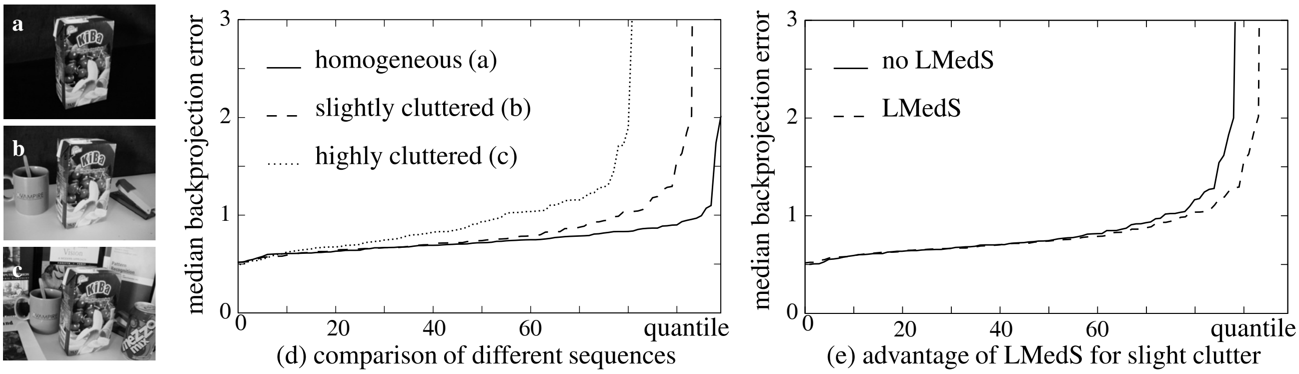


Figure 3: Three images of a video sequence in which a toy elk was tracked with the adaptive 3-D hyperplane tracker. It can clearly be seen that the points of the model are placed very accurately on the object, even if the appearance changes drastically. Although the camera and the object are moved at the same time, the object is tracked successfully.



(d) comparison of different sequences

(e) advantage of LMedS for slight clutter

Figure 4: Detection results for three test image sequences where a package of juice has to be detected in front of a homogeneous (a), slightly cluttered (b), and highly cluttered background. The median backprojection error of the model points are calculated for every image and sorted in ascending order. The first graph (d) shows the quality of the pose estimation using the LMedS algorithm for the three different background types. The second graph (e) illustrates the median backprojection error of the "slightly cluttered" sequence (b) with and without application of the LMedS algorithm.

References

[1] M. Black and A. Jepson. Eigen tracking: Robust matching and tracking of articulated objects using a view-based representation. In B. Buxton and R. Cipolla, editors, *Computer Vision - ECCV'96, 4th European Conference on Computer Vision*, pages 329–342, Cambridge, UK, 1996. Springer, Heidelberg.

[2] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003.

[3] D. Dementhon and L. S. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15(1–2):123–155, 1989.

[4] Ch. Gräßl, T. Zinßer, and H. Niemann. Illumination insensitive template matching with hyperplanes. In *Pattern Recognition, 25th DAGM Symposium*, pages 273–280, Magdeburg, 2003. Springer, Heidelberg.

[5] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, 2000.

[7] B. Heigl. *Plenoptic Scene Modeling from Uncalibrated Image Sequences*. Ibidem-Verlag, Stuttgart, January 2004.

[8] F. Jurie and M. Dhome. Real-time 3D template matching. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recongition*, volume 1, pages 791–797, Kauai, 2001. IEEE Computer Society Press, Washington.

[9] F. Jurie and M. Dhome. Hyperplane approach for template matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):996–1000, 2002.

[10] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th International Conference on Computer Vision*, volume 2, pages 1150–1157, Corfu, 1999. IEEE Computer Society, Washington.

[11] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *Proceedings of the European Conference on Computer Vision*, pages 661–675, Copenhagen, 2002. Springer, Heidelberg.

[12] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, New York, 1987.

[13] W. Sepp and G. Hirzinger. Real-time texture-based 3-D tracking. In *Pattern Recognition, 25th DAGM Symposium*, pages 330–337, Magdeburg, 2003. Springer, Heidelberg.

[14] M. Wagner, U. Labsik, and G. Greiner. Repairing non-manifold triangle meshes using simulated annealing. *International Journal on Shape Modeling*, 9(2):137–153, 2003.

[15] M. Zobel, M. Fritz, and I. Scholz. Object tracking and pose estimation using light-field object models. In *Vision, Modeling, and Visualization 2002*, pages 371–378, Erlangen, 2002. Aka / IOS Press, Berlin.