B. Girod, M. Magnor, and H.-P. Seidel: Vision, Modeling and Visualization 2004, Stanford, 2004, Aka / IOS Press, Berlin, Amsterdam

# A Probabilistic Model-Based Template Matching Approach for Robust Object Tracking in Real-Time

Christoph Gräßl*, Timo Zinßer*, Heinrich Niemann

Chair for Pattern Recognition
University of Erlangen-Nuremberg
Martensstraße 3, 91058 Erlangen, Germany
Email: {graessl, zinsser}@informatik.uni-erlangen.de

## Abstract

In recent years, template matching approaches for object tracking in real-time have become more and more popular, mainly due to the increase in available computational power and the advent of very efficient algorithms. Particularly, data-driven methods based on first order approximations have shown very promising results. If the object to be tracked is known, a model-based tracking algorithm is preferable, because available knowledge of the appearence of the object from different views can be used to improve the robustness of the tracking. In this paper, we enhance the well-known hyperplane tracker with a probabilistic tracking framework using the CONDENSATION algorithm, which is noted for its robustness and efficiency. Furthermore, we put forward a subspace method for improving the tracker's robustness against illumination variations. We prove the efficiency of our proposed methods with experiments on video sequences of real scenes.

## 1 Introduction

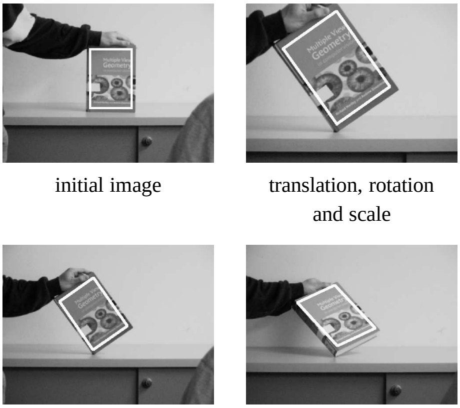
Object tracking is used as a low-level component of many different vison-based applications like medical imaging [10] and video surveillance [15]. The main aim of visual object tracking is to estimate the position of an object in a sequence of consecutive images. As a precondition it is assumed that the images are captured in short time intervals.

A well established route toward efficient tracking is to use color-histogram features for matching a region in the current frame with a reference region. For this optimization problem, an analytical approach using the meanshift algorithm has been presented by [3] and a probabilistic approach using the *CONDENSATION* algorithm has been developed by [14]. Both approaches assume that the histogram of the object remains constant in all images. However, this precondition does not hold in case of strong appearence changes, which can be caused by relative motion between object and camera, for example. These approaches also suffer from their inability to estimate the object's rotation.

On the other hand, the template matching approaches presented by Hager [6] and Jurie [8] are very promising because of their accuracy and their capability to compute additional motion parameters in the 2-D image plane, like rotation or perspective distortion. Both approaches use a first order approximation, which results in high computational efficiency and facilitates template matching in real-time. For improving the robustness of the tracking algorithm, a linear illumination model to cope with strong illumination changes can be used [6, 4]. Due to their data-driven nature, these approaches cannot handle strong viewpoint changes.

Jurie captured 250 views of the object on a hemisphere, and improved his approach with the resulting appearence-based object model [9]. He trains a separate hyperplane tracker for every view, and during tracking, chooses the motion parameters estimated by the tracker yielding the lowest error. To improve the speed, he uses only a fixed number of trackers, which were trained for views similar to that of the best tracker of the previous frame. Although a method for illumination compensation by estimating an affine transformation of the gray-level intensities is proposed, our experiments show that a change of the relative position of the light source

*This work was funded by the European Commission 5th IST Programme - Project VAMPIRE. Only the authors are responsible for the content.



initial image | translation, rotation and scale | translation and rotation | affi ne motion

Figure 1: Three examples for tracking with different motion parameterizations. The reference template was taken from the initial image.

leads to bad results if the object is not planar.

In this paper, we use the idea of [9] to build an object model by training a separate hyperplane tracker for each of a large number of viewpoints. But instead of using a fixed number of trackers for each frame, we employ a particle filter approach based on the CONDENSATION algorithm [7] for the selection of the tracker. Particle filters are very popular for object tracking because they have proven to be very robust and efficient.

In our work, there are two main advantages of using a particle filter approach. If the prediction of a viewpoint is very certain, the particles are concentrating on a small neighborhood of possible viewpoints. Thus only a small number of hypotheses has to be taken into account, which saves computational resources. On the other hand, if the predictions are very uncertain, especially in the case of ambiguities, the fixed number of trackers as proposed in [9] may be too small. The properties of the particle filter ensure that even viewpoints which are far away from the currently estimated viewpoint are taken into account if their probability is high enough. Additionally, we show how to use the silhouette of the object for a very precise segmentation of the object. In contrast to this, [9] uses only an elliptical region.

Furthermore, we will detail how to use a linear subspace method (proposed by [1]) to model image variations due to illumination changes. Some other methods for dealing with variations of the lighting conditions have been shown in [4], but these approaches only work well with planar object. Throughout, we present experimental results performed on live video sequences demonstrating the performance of our methods.

The paper is structured as follows. In Sect. 2, we will give a review of Jurie's 2-D template matching algorithm. This algorithm is used in a probabilistic model-based framework, which is presented in Sect. 3. Sect. 4 details an subspace approach for dealing with illumination variations. The experiments we have conducted are documented in Sect. 5. After a summary of our work, possible future extensions are discussed in Sect. 6.

## 2 2-D Template Matching

Our method for model-based object tracking is based on a data-driven 2-D template matching technique, the hyperplane approach from Frederic Jurie [8]. Template matching algorithms for data-driven tracking work on a sequence of gray-level images $\boldsymbol{f}_t$ captured at time $t$. We have shown that using color information can lead to better accuracy [5], but for the sake of simplicity we will only use gray-level data throughout the rest of this paper.

The object to be tracked is specified by a *reference template* in the *reference image* $\boldsymbol{f}_{t_0}$. The reference template is defined by vector $\boldsymbol{r} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_r})^T$, which contains the 2-D coordinates of the template points. The gray-level intensity of a point $\boldsymbol{x}_i = (x_i, y_i)^T$ at time $t$ is given by $f(\boldsymbol{x}, t)$. Consequently, 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 $t$ can be modelled by $\boldsymbol{r}_t = \boldsymbol{g}\left(\boldsymbol{r}, \boldsymbol{\mu}(t)\right)$, where vector $\boldsymbol{\mu}(t) = (\mu_1(t), \mu_2(t), \ldots, \mu_n(t))^T$ contains the *motion parameters*. Sample images of tracking with different motion parametrizations are shown in Fig. 1. Template matching can now be described as computing the motion parameters $\boldsymbol{\mu}(t)$ that minimize the least-square intensity difference between the reference template and the current template:

$$\boldsymbol{\mu}(t) = \underset{\boldsymbol{\mu}}{\operatorname{argmin}} \left\| \boldsymbol{f}\left(\boldsymbol{r}, t_0\right) - \boldsymbol{f}\left(\boldsymbol{g}\left(\boldsymbol{r}, \boldsymbol{\mu}(t)\right), t\right) \right\|_2 . \tag{1}$$

Non-linear minimization of Eq. (1) involves extremely high computational cost [2], which is disadvantageous for real-time applications. It is more efficient to approximate $\boldsymbol{\mu}$ by a linear system

$$\begin{aligned}\boldsymbol{\mu}(t+1) &= \boldsymbol{\mu}(t) + \boldsymbol{A}(t+1)\boldsymbol{e}_t \\ \boldsymbol{e}_t &= \boldsymbol{f}(\boldsymbol{r}, t_0) - \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{r}, \boldsymbol{\mu}(t)), t+1)\end{aligned} \tag{2}$$

as presented in [6, 8]. There are two approaches for computing matrix $\boldsymbol{A}(t)$ from equation (2). Hager and Belhumeur [6] propose using a Taylor approximation. The hyperplane approach presented in [8] acquires matrix $\boldsymbol{A}$ by a least-square estimation. In the latter approach, matrix $\boldsymbol{A}$ can be expressed independent from time $t$, but has to be computed in a separate training step when the initial image and the reference template are available. As the hyperplane approach has a superior basin of convergence, we will use it throughout the rest of this paper.

Eq. (2) clearly illustrates that appearence changes due to varying illumination lead to estimation errors, because the motion parameters directly depend on the gray-level intensities. [4] compared different methods for dealing with these illumination variations and suggests the use of a linear illumination model, which compensates for changes of brightness and contrast.

## 3 Probabilistic Viewpoint Estimation

After the review of Jurie's template matching algorithm for data-driven object tracking, we show how this approach can be used in a model-based framework. Our goal is not only to estimate the object's 2-D position, scale and in-plane rotation, but also to determine the current viewpoint of the object, which facilitates a highly accurate segmentation of the object from the background.

### 3.1 Object Model

Usually, sample images of the object from different viewpoints are needed for model-based algorithms. For our experiments, we used a *Santa Claus* made of clay and put it on a turn table, where sample images were captured with a camera mounted on a robot arm (illustrated in Fig. 2).

The main idea of model-based hyperplane tracking [9] is to train an update matrix $\boldsymbol{A}_{\boldsymbol{\nu}}$ for the reference template $\boldsymbol{r}_{\boldsymbol{\nu}}$ of every sample image $\boldsymbol{f}_{\boldsymbol{\nu}}$ with viewpoint $\boldsymbol{\nu} = (\nu_T, \nu_R)^T$, where $\nu_T$ is the turn table angle and $\nu_R$ is the angle of the robot arm. For tracking at time $t$, each matrix $\boldsymbol{A}_{\boldsymbol{\nu}}$ is used to estimate the motion parameter vector

$$\begin{aligned}\boldsymbol{\mu}_{\boldsymbol{\nu}}(t+1) &= \boldsymbol{\mu}(t) + \boldsymbol{A}_{\boldsymbol{\nu}}(t+1)\boldsymbol{e}_t(\boldsymbol{\nu}) \\ \boldsymbol{e}_t(\boldsymbol{\nu}) &= \boldsymbol{f}(\boldsymbol{r}_{\boldsymbol{\nu}}, t_0) - \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{r}_{\boldsymbol{\nu}}, \boldsymbol{\mu}(t)), t+1)\,.\end{aligned} \tag{3}$$

Thus, the motion parameter of the motion in the image plane and the most similar viewpoint can be retrieved by

$$\begin{aligned}&(\boldsymbol{\mu}(t), \boldsymbol{\nu}(t)) = \\ &\operatorname*{argmin}_{\boldsymbol{\mu}_{\boldsymbol{\nu}}, \boldsymbol{\nu}} \left\| \boldsymbol{f}(\boldsymbol{r}_{\boldsymbol{\nu}}, t_0) - \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{r}_{\boldsymbol{\nu}}, \boldsymbol{\mu}_{\boldsymbol{\nu}}(t)), t) \right\|_2 ,\end{aligned} \tag{4}$$

where $\boldsymbol{g}(\boldsymbol{r}_{\boldsymbol{\nu}}, \boldsymbol{\mu}_{\boldsymbol{\nu}})$ transforms the reference template $\boldsymbol{r}_{\boldsymbol{\nu}}$ of sample image $\boldsymbol{f}_{\boldsymbol{\nu}}$. Although [8, 4] use a motion parameterization which allows the estimation of perspective distortions, a parameterization which allows for 2-D translation, in-plane rotation, and scale is absolutely sufficient. Thus, the transformation of one point $\boldsymbol{x}$ of vector $\boldsymbol{r}_{\boldsymbol{\nu}}$ is given by

$$\begin{aligned}&\boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\mu}) = \\ &\begin{pmatrix} \frac{\cos\mu_3}{\mu_4} & -\frac{\sin\mu_3}{\mu_4} \\ \frac{\sin\mu_3}{\mu_4} & \frac{\cos\mu_3}{\mu_4} \end{pmatrix} \boldsymbol{x} + \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}.\end{aligned} \tag{5}$$

In contrast to the work of [9], we use a separate template, which is exactly aligned to the object's appearence, for every viewpoint of the sample set. This is not difficult as the sample images can be taken in a controlled environment (i.e. black background and good lighting conditions) and object segmentation can be done by simple thresholding. The points of $\boldsymbol{r}_{\boldsymbol{\nu}}$ are restricted to be placed inside the object and enforced to have a certain distance to the border, which increases the robustness of the tracking. In addition, we propose to extract $N_S$ points on the border of the object to retain a silhouette region $\boldsymbol{w}_{\boldsymbol{\nu}} = (\boldsymbol{x}'_1, \boldsymbol{x}'_2, \ldots, \boldsymbol{x}'_{N_S})^T$ of the object, which enables a very precise segmentation of the object and also supports a meaningful visualization of the tracking result. The silhouette of the object with respect to the current motion parameter vector and viewpoint is given by

$$\boldsymbol{w}(t) = \boldsymbol{g}(\boldsymbol{w}_{\boldsymbol{\nu}(t)}, \boldsymbol{\mu}(t)). \tag{6}$$

### 3.2 Probabilistic Framework

In principle, the viewpoint selection can easily be performed by testing the whole sample set and using the motion estimation of the matrix $\boldsymbol{A}_{\boldsymbol{\nu}}$ that

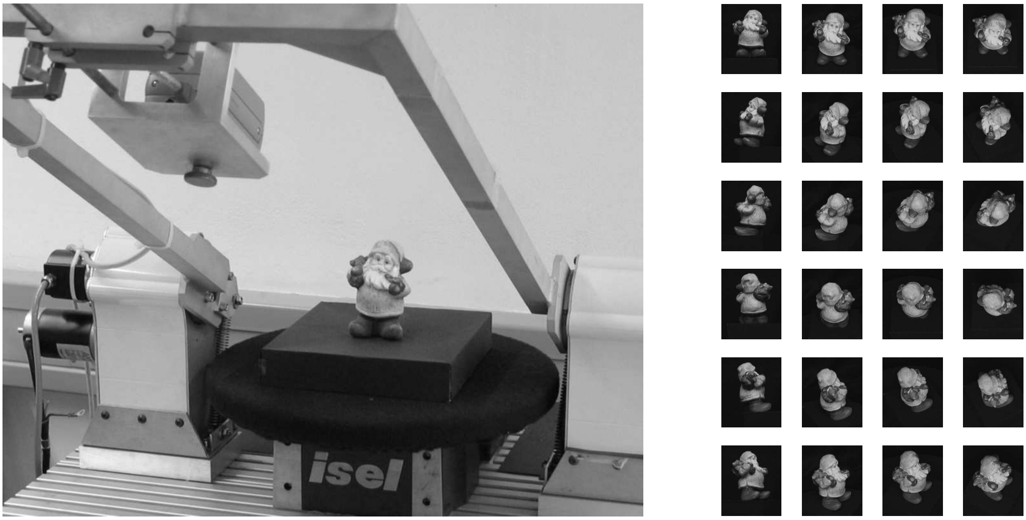


Figure 2: Left: A turntable and a robotarm for capturing object images on a hemisphere around the object. Right: Different object images from a *Santa Claus* made of clay.

gives the best result (cf. Eq. (1)). As testing all possible hypotheses takes a lot of computation time, [9] uses only a neighborhood of sample images whose viewpoint is similar to the viewpoint estimated for the previous frame. This approach works well under the assumptions that changes of viewpoint do not occur too rapidly and that the viewpoint estimation in the previous frame was correct. The disadvantage is that too small a set of hypotheses will lead to bad estimation results, especially if some of the viewpoint estimations are wrong or very uncertain (e.g. in case of ambiguities). But on the other hand, too big a set will require an unnecessary amount of computational resources, especially if the hypothesis are very certain. We propose to use a dynamic framework based on the particle filter approach, the CONDENSATION algorithm [7], as it has proven to be a very robust technique during its widespread use in the area of object tracking.

Particle filters for tracking generally estimate the location of an object in which the posterior density $p(\boldsymbol{s}_t|\boldsymbol{O}_t)$ and the observation density $p(\boldsymbol{O}_t|\boldsymbol{s}_t)$ are often of non-Gaussian nature. The vector $\boldsymbol{O}_t$ denotes all observations $\boldsymbol{o}_t, \boldsymbol{o}_{t-1}, \ldots, \boldsymbol{o}_{t_0}$. Different representation of vector $\boldsymbol{s}_t$ containing position, scale, rotation, speed, and/or acceleration are presented in [14, 13, 16]. In our case, the state $\boldsymbol{s}_t$ contains the two parameters of the viewpoint $\boldsymbol{\nu}$, as all other relevant parameters are estimated by the hyperplane tracker.

The principle of particle filtering is to approximate the probability distribution of the object's state by a weighted particle set $\boldsymbol{C} = \{(\boldsymbol{s}_i, \pi_i)\,|i = 1, 2, \ldots, N\}$, where each particle consists of a hypothetical state of the object and a discrete probability $\pi_i$ for this hypothesis with $\sum_i \pi_i = 1$.

One cycle of the CONDENSATION algorithm is presented in Fig. 3. At first, a sample set is calculated using the estimated probability distribution from the last step. In the *drift*-phase, all particles change in a deterministic way. For object tracking, the state transition is used for a linear prediction of the position of the object. In the *diffuse*-phase, Gaussian noise is added to all particles in order to model the uncertainty. A second reason is that states with high probability are usually represented by several particles, which are scattered after the diffusion. After this, the new states are measured using the current image, new weights are assigned to the particles, and a new distribution is represented by the new particle set.

One possibility for measuring the states is to compare color histograms of the reference window with the window represented by the particle [14]. Another possibility is to directly compare the in-

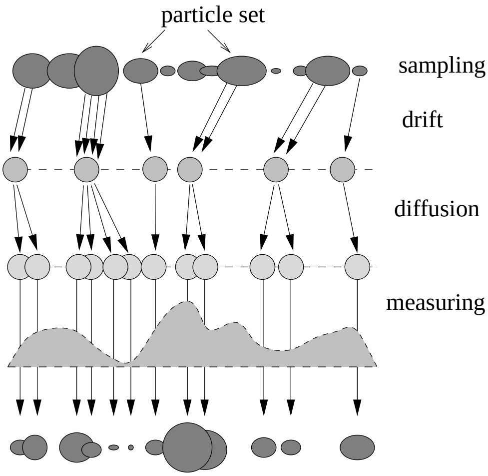



Figure 3: The CONDENSATION algorithm [7]

tensities in the reference template with the intensities of the template represented by the particle using correlation-based techniques [12, 16]. We are using the mean square difference $e_i$ of the intensities which has been optimized according to Eq. 4. Then, we compute a probability

$$\pi_i = \tau e^{-\sigma e_i} \tag{7}$$

for every particle, where $\tau$ is a normalization coefficient for ensuring $\sum_i \pi_i = 1$. In our experiments, we found out that the strictness parameter $\sigma = 0.0005$ leads to good results. There are different techniques for retrieving the maximum of the approximation of the probability distribution of the object's state. Usually, the maximum is estimated by an optimization of a parzen density distribution, but as this technique is computationally expensive, we propose to use the state of the particle with the highest probability.

## 4 Illumination Variations

It was already mentioned in chapter 2 that appearence variations due to illumination changes must not be ignored, because the estimation of motion directly depends on gray-level intensities. Assuming a planar surface, the intensity of a pixel can be adjusted with a linear model

$$f_{\text{new}}(\boldsymbol{x}) = \beta f(\boldsymbol{x}) + \gamma \quad \forall \boldsymbol{x} \in \boldsymbol{r} \tag{8}$$

Figure 4: The upper left image can be transformed to the upper right image by an intensity distribution normalization, because the images differ only in brightness and contrast. However, the lower images cannot be transformed into one another in this way, because the differences are caused by a change of the position of the light source.

with illumination compensation parameters $\beta$ and $\gamma$, which represent variations of contrast an brightness. Experimental results in [4] showed that these parameter can be retrieved by a normalization of distributions of the template intensities. This model becomes inadequate when considering non-planar surfaces and a changing relative position of the light source, which is illustrated in Fig. 4.

For dealing with such illumination changes, we propose to use a subspace method based on the work of [1], which has already been successfully adapted to a tracking approach [6]. It was shown that a small number (at least three) of images of the object, all captured from the same viewpoint but under different illumination conditions, are sufficient to reconstruct object images under different illumination conditions, as far as a lambertian reflection model, monochrom images, and a convex object can be assumed.

The key idea is to compute an orthogonal base using images of the object, all captured from the same viewpoint but under different illumination, with a singular value decomposition. As illustrated in Fig 5, for our experiments we used six images for every viewpoint under different illumination conditions and computed $N_B = 4$ basis images $\boldsymbol{b}_i, i =$

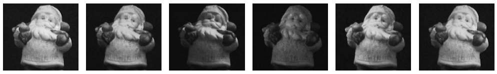


Figure 5: Images of the Santa Claus figurine captured from one viewpoint but under different illumination conditions, which were used for estimating the basis images [1]

$1, \ldots, N_B$. Instead of the whole image, only the intensities of the template are used. For simplicity, we demonstrate this technique for one viewpoint. An image can be reconstructed with the linear model

$$\hat{\boldsymbol{f}} = \sum_{i=1}^{N_B} \lambda_i \boldsymbol{b}_i, \tag{9}$$

where $\lambda_i$ are the corresponding illumination parameters for basis image $\boldsymbol{b}_i$. The estimation of the illumination parameters of an image $\boldsymbol{f}$ is given by the solution of the overdetermined system

$$\boldsymbol{f} = \boldsymbol{B\lambda}, \tag{10}$$
$$\boldsymbol{\lambda} = \boldsymbol{B}^{\#}\boldsymbol{f}, \tag{11}$$

where matrix $\boldsymbol{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{N_B})$, vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_{N_B})^T$, and $\boldsymbol{B}^{\#}$ denotes the pseudoinverse of matrix $\boldsymbol{B}$.

During tracking, an illumination change can be compensated by using the following equation derived from Eq. (2)

$$\hat{\boldsymbol{\mu}}(t+1) = \hat{\boldsymbol{\mu}}(t) + \boldsymbol{A}(t+1)(\boldsymbol{e}_t - \boldsymbol{B\lambda}(t-1)), \tag{12}$$

where the basis images only contain values for the corresponding template, not for the whole image. The illumination parameters for time step $t$ can be computed by

$$\boldsymbol{\lambda}(t) = \boldsymbol{B}^{\#} \left( \boldsymbol{f}\left(\boldsymbol{r}, t_0\right) - \boldsymbol{f}\left(\boldsymbol{g}\left(\boldsymbol{r}, \boldsymbol{\mu}(t)\right), t\right)\right), \tag{13}$$

which is computationally efficient, because $\boldsymbol{B}^{\#}$ only has to be estimated once in an offline stage. The efficiency of this method can be improved by appending a unit vecor to $\boldsymbol{B}$ to model changes of brightness [6].

## 5 Experimental Results

The following experiments with image sequences of real scenes demonstrate the efficiency of our proposed methods. We used the setup presented in

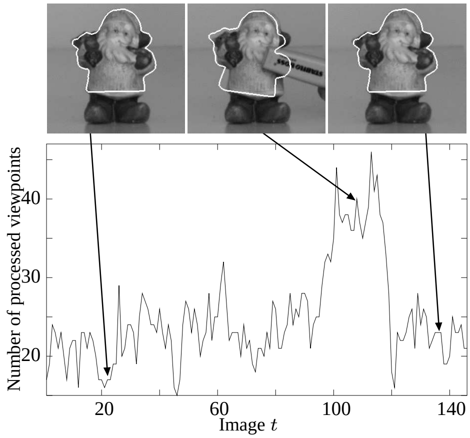



Figure 6: If the estimation of the viewpoint is uncertain (e.g. caused by partial occlusion), the number of processed viewpoints will increase

Fig. 2 to capture 5400 images of one object for 900 different poses and 6 different illumination conditions with a Sony DFW-VL 500 firewire camera at a resolution of $640 \times 480$ pixels. Masks for the segmentation of the object were created for 900 images captured with one fixed illumination by thresholding. All mask images, which were applied to all images of the object, were manually verified and corrected in case of segmentation errors.

In the first experiment, we tested the quality of the particle filter in case of uncertainty. As discussed in Sect. 3, most of the particles should concentrate on one viewpoint if the estimation is very certain (i.e. the error minimized in Eq. (4) is small). This improves the computation speed of the system, because only a small number of viewpoints has to be processed. In our experimental setup, we tracked the Santa Claus using the proposed particle filter approach with 100 particles. In the diffuse phase (cf. Fig. 3), we added Gaussian noise with a variance of 5 degrees. The sample video sequence with 148 images shows an object that becomes partially occluded by a pen, which increases the uncertainty of the viewpoint estimation. As a result, the particles immediately spread out, which is illustrated in Fig. 6.

In Sect. 4, we presented an approach for reducing the illumination sensitivity of the hyperplane approach for model-based object tracking by using

Figure 7: The upper images show results of tracking without basis images and the lower images show results using basis images. The benefits of the basis image approach is clearly visible.

illumination basis images. In our experiments, we created four basis images from six images of the object for every viewpoint . For testing our methods, we compared the basis image approach with tracking without basis images. We illuminated the object from different directions and also shadowed the object. The tracker was trained with a region size of $N_r = 150$ pixels and 2000 random warpings for the estimation of matrix $\boldsymbol{A}$. A coarse-to-fine strategy with three 2-D trackers was employed to improve the estimation accuracy as proposed in [8]. In Fig. 7, some sample images of the described video sequence are presented. It is clearly visible that using illumination basis images leads to better tracking results than using no basis images. Some sample images of another long tracking sequence with 863 frames are shown in Fig. 8.

It is obvious that a template consisting of a high number of points leads to a higher accuracy than a low number, but for many tracking applications, the computation time of the algorithms is an important issue. In order to illustrate the interdependency of the region size $N_r$ and the tracking time, we present the according values for different configurations in Tab. 1. The values were acquired on a Pentium 4

| basis images | $N_r = 150$ | $N_r = 300$ | $N_r = 700$ |
|---|---|---|---|
| applied | 267 | 325 | XXX |
| not applied | 260 | 320 | XXX |

Table 1: Computation time in milliseconds for one frame depending on the region size $N_r$ of the template

PC with 2.4GHz and 1GB main memory.

## 6 Conclusion and Outlook

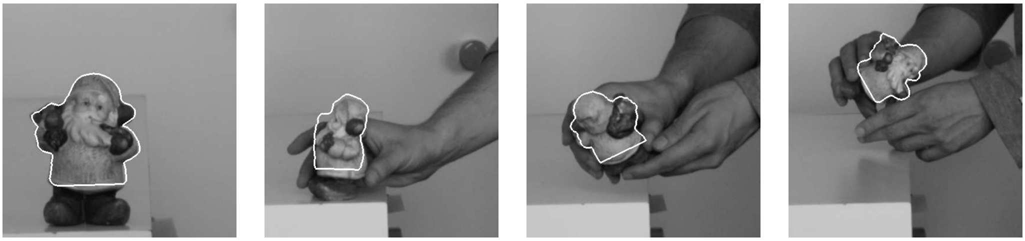
We presented a probabilistic method for model-based object tracking based on the CONDENSATION algorithm and the hyperplane approach. For reducing the illumination sensitivity, we used a linear subspace method for reconstructing images under variable lighting conditions using illumination basis images. Experiments conducted with real image sequences prove the efficiency of our proposed methods. We showed that our probabilistic approach flexibly adapts to the uncertainty of the estimations, which improves both computation time and robustness.

One drawback of this approach is the laborious initialization of the tracking system. In the first frame of a sequence, the best fitting model is estimated with an exhaustive search using a template matching technique. This is computationally expensive and not very robust. For dealing with this problem, local features like Lowe's SIFT key points [11] seem to be very promising. Using this methods also enables a reinitialization when the tracking system loses the object. Another improvement would be the use of a more robust technique for measuring the states of the particle filter.

## References


[1] P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible lighting conditions. In *IEEE Confernce on Computer Vision and Pattern Recognition*, pages 270–277, San Francisco, 1996. IEEE Computer Society Press, Washington.

[2] M. Betke and N. Makris. Fast object recognition in noisy images using simulated annealing. Technical Memo AIM-1510, Mas-



frame 10 frame 110 frame 2200 frame 800

Figure 8: Images of a long tracking sequence with 863 frames. The tracking system successfully estimated the position, rotation and scale of the object. In frame 110, a wrong model was estimated, which caused some misalignment of the template and the object, but was automatically corrected in subsequent frames.


sachusetts Institute of Technology, Artificial Intelligence Laboratory, January 25, 1995.

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

[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] Ch. Gräßl, T. Zinßer, and H. Niemann. Efficient Hyperplabe Tracking by Intelligent Region Selection. In *6th IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 51–55, Lake Tahoe, 2004. IEEE Computer Society Press, Washington.

[6] G.D. Hager and P.N. Belhumeur. Efficient Region Tracking with Parametric Models of Geometry and Illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, 1998.

[7] M. Isard and A. Blake. CONDENSATION—conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.

[8] F. Jurie and M. Dhome. Hyperplane Approach for Template Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):996–1000, 2002.

[9] F. Jurie and M. Dhome. Real time tracking of 3d objects : a robust approach. *Pattern Recognition*, 35(2):317–328, 2002.

[10] D.J. Kang and I.S. Kweon. Fast and Stable Snake Algorithm for Medical Images. *Pattern Recognition Letters*, 20(5):507–512, 5 1999.

[11] D. G. 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.

[12] J. Martin and J. Crowley. Experimental comparison of correlation techniques. In *3rd International Symposium on Intelligent Robotic Systems - SIRS'95*, pages 287–294, Pisa, 1995.

[13] K. Nummiaro, E. Koller-Meier, and L. Van Gool. Object tracking with an adaptive color-based particle filter. In *Pattern Recognition, 24th DAGM Symposium*, pages 353–360, Zurich, 2003. Springer, Heidelberg.

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

[15] F. Ziliani and J. Reichel. Effective Integration of Object Tracking in a Video Coding Scheme for Multisensor Surveillance Systems. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 521–524, Rochester, USA, September 2002. IEEE Computer Society Press.

[16] 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, Germany, 2002. Infix.