Object Tracking and Pose Estimation Using Light-Field Object Models

Matthias Zobel*, Mario Fritz, Ingo Scholz

University of Erlangen, Department of Computer Science
Chair for Pattern Recognition
Martensstraße 3, 91055 Erlangen, Germany
Email: info@immd5.informatik.uni-erlangen.de

Abstract

Geometric object models have been widely used for visual object tracking. In this contribution we present particle filter based object tracking with pose estimation using an appearance based light-field object model. A light-field is an image-based object representation which can be used to render a photo realistic view of an arbitrarily shaped object from arbitrary viewpoints. It is shown how light-field object models can be generated and utilized. Furthermore, we show how these models fit into the probabilistic framework of dynamic state estimation by defining an appropriate likelihood distribution from an image similarity metric. Finally, we present results and accuracy evaluations from tracking experiments of different objects.

1 Introduction

Object tracking is the task of sequentially estimating the internal, unknown state of a moving object from a sequence of observations. Model based object tracking exploits the fact that information about the object is available in form of a model, from which it is possible to predict what observations should be expected, if a certain state of the object is assumed. This predicted observation is compared to the actual observation and the estimated state could be updated accordingly. In 3-D object tracking, the state of the object normally is a vector that contains the 3-D position of the object, its velocity and acceleration. Additionally, sometimes the pose of the object is also of interest and is therefore, appropriately parameterized, included in the state vector. In this contribution we present results on the use of appearance based light-field object models for 3-D object tracking with pose estimation.

The literature proposes many different kinds of object models to use for object tracking. The classical and most common models are geometric object models. These models mainly describe the shape of an object by a set of lines, curves, surfaces, etc. For example, a 3-D CAD model belongs to this class of models. Geometric object models are well suited to derive and predict the outline of an object given its 3-D position and pose, i.e., when it is seen from a certain viewpoint. Hence, those models are often successfully deployed in contour tracking applications based on segmentation information (edges, corners, etc.) [10, 17] . One drawback of those models is, that geometric information on the object has to be available in order to generate the model. This data might be difficult to achieve, or not even available. Additionally, there are objects showing such a geometric complexity that their shape cannot be described appropriately. Hence, objects either have to be shaped relatively simple (mostly, polyhedral objects are preferred, e.g., [12]) or can be approximated by simple shaped geometric primitives, like cuboids or ellipsoids.

Instead of comparing segmented edges with predicted ones, a different approach to object tracking is to predict how the object (or parts of it) should look like from a certain viewpoint and to compare this to the actual observation. Therefore, object models are necessary that provide information from which any view of the object can be rendered. Those models could either be synthesized and approximate the appearance of the object by means of computer graphics [14], or the model could also be built from several real images of an object. A well known appearance based object model of this kind is the view-based eigenspace representation of an object (e.g., [13]), which approximates the ob-

*This work was supported by the “Deutsche Forschungsgemeinschaft” under grant SFB603/TP B2 and C2. Only the authors are responsible for the content.

ject's appearance by a linear subspace. In [2] this object model is used for 2-D tracking (i.e., in the image plane) of a cylindrically shaped object. In [16] a similar approach is used to build generative appearance models of human limbs that were used for tracking human motion in 3-D.

In this contribution we use so called *light-fields* [4, 11] to capture and store the appearance of an object. A light-field is a view-based representation of an object from which photo-realistic images of the object from arbitrary viewpoints can be rendered in real-time by interpolating between known views. A light-field can be easily generated for arbitrarily shaped objects without any a priori knowledge about the object's geometry (details are given in Section 2). The applicability of light-field object models is widespread, for example, they were used for training a statistical classifier [5], or even for vision based robot navigation [6].

The work presented here uses light-field object models for visual 3-D tracking and pose estimation. We show the applicability of this object model in the framework of dynamic state estimation with particle filters. We present evaluations of the estimation accuracy for different objects and trajectories under controlled conditions.

This contribution is organized as follows. In the next section we first give an introduction to light-field object models. We describe the task of model generation for the objects we have used. Afterwards, we state the task of 3-D object tracking in the probabilistic framework of Bayesian state estimation. We shortly review the particle filter approach to sequential state estimation that has been used for the tracking experiments throughout this paper. In this context of particle filters a discussion follows of the correlation based image similarity metric we have chosen to define a likelihood probability density function (pdf) for weighting the individual particles. The experiments we conducted as well as the obtained results are finally presented in Section 4.

2 Light-Fields

In contrast to model-based rendering light-fields use camera images of real scenes and objects to render new views of these scenes and objects from arbitrary viewpoints, without having an explicit geometric model. The advantages of this technique are that objects and scenes can be reproduced in photo-realistic quality including even effects of specularity, and that images can be rendered quickly despite a complex geometry of the scene.

The idea behind light-fields is that each image can be considered as a bundle of light rays, where each ray is emitted from a certain point on the scene surface in a certain direction towards the center of projection of the camera. Thus the light-field consists of a collection of light rays, where every surface point has been sampled from a number of different directions. An image from a new and previously unsampled viewpoint is then constructed by looking for the closest known rays to each of the light rays in the image and interpolating their intensity over them.

Storing a light-field is a question of parameterizing the known light rays. In the original papers [4, 11] light-fields have been parameterized using two planes, although other parameterizations have been proposed (see [15] for references). Each plane consists of a regular grid in world space, where one grid holds the positions of the cameras of each view, and the other is usually set to the distance of the image planes.

The problem with this parameterization is that either the cameras have to be set to these grid positions while sampling the scene, which can be done using a sophisticated mechanical setup, or that the images are warped after recording such that they fit into the two-plane grid, introducing a loss in quality. Therefore the light-fields used here are stored as so-called free form light-fields as described in [7]. They are a generalization of the two-plane approach as they allow cameras to be situated on an arbitrary but convex mesh around the scene.

This configuration is very suitable for the object representation here, as the camera used for recording the images of the objects was mounted on a moving robot arm above a turntable so that the camera positions are situated on a hemisphere around the object. By using the hardware acceleration schemes described in [15] it is possible to render several images of the object per second using a free form light-field.

Another way to improve the results of light-field rendering is to use depth information, as it allows to identify the exact location on the surface of the scene where it is intersected by a certain light ray and to find the corresponding known light rays necessary for interpolation. Nevertheless in the experiments described later the scene depth for each view is approximated by a plane through the object, resulting at some points in a superpositioning of several images, also referred to as ghosting artifacts.

2.1 Light-Fields as Object Models

In the remainder of this section we describe the generation of the light-field object models. In order to record the images of each object we use the setup shown in Figure 1. A camera is mounted on a robot arm above a turntable which both can be moved (tilted or rotated respectively) by step motors.

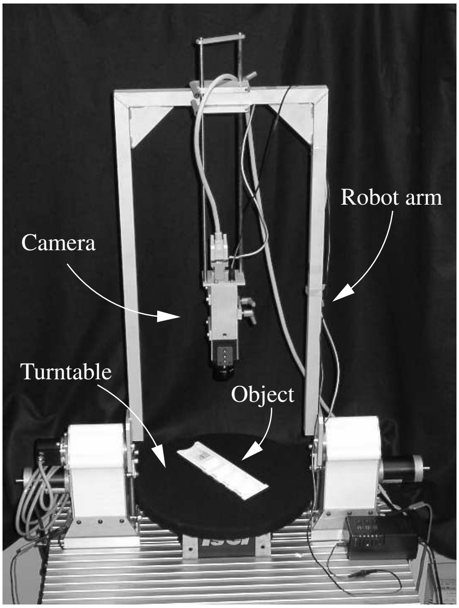

Figure 1: Turntable and robot arm assembly for light-field sampling.

Using this setup the cameras of the samples for reconstructing the light-field are placed on a hemisphere around the object as shown in Figure 2. To reach a uniform sampling, the number of samples taken for each rotation was reduced the higher up the camera position was on the hemisphere. The intrinsic camera parameters were calibrated automatically from the sample images using a procedure similar to [9]. For the focal length an initial value was used which is close to the real value to reach more stable results. In general this initialization can

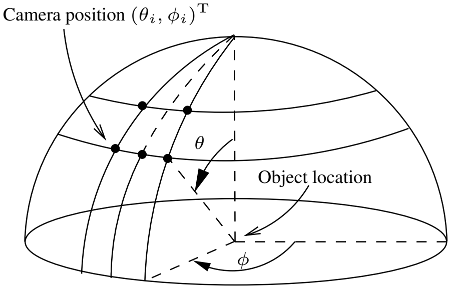

Figure 2: Geometric configuration for light-field sampling.

be an arbitrary value. The camera positions were then computed using the angles θ and ϕ of the robot arm and the turntable.

While it is possible to use automatic camera calibration for the extrinsic camera parameters as well, in many cases it does not lead to satisfying results. Especially on objects with specular surface or mostly uniform texture the feature tracking algorithm often fails to generate enough point correspondences for calibrating the whole hemisphere of images. Therefore we decided against this automatic calibration, as otherwise the choice of objects would be reduced too much. Calibration would also yield depth information for the point correspondences as a byproduct, which is thus unavailable. Instead we approximate depth information using planes at the mean distance of the object from the cameras. Each plane is parallel to the image plane of the corresponding camera, i.e., a local depth map as in [7] without depth correction [4].

Once the light-field of the object model is created, arbitrary views can be rendered from it. The virtual camera coordinates are parameterized using the two angles on the hemisphere at a standard distance from its center. Variations in size of the object to track may originate from a changing distance of the camera or a different focal length. Due to constraints in the hardware-accelerated free form light-field renderer we use, we model size variations by adjusting the field-of-view of the camera, which corresponds to a change in focal length.

3 Object Tracking by Particle Filters

Visual 3-D Object tracking is in general seen as a sequential state estimation problem. The unknown, varying state of the object at a certain timestep, e.g., its current 3-D position, its velocity, or its pose, has to be estimated from a sequence of observations made by one or more cameras up to this time. Formulating the problem in a Bayesian framework, object tracking means the sequential estimation of the a posteriori pdf $p(\boldsymbol{q}_t|\langle \boldsymbol{o}\rangle_t)$ at time t of the states $\boldsymbol{q}_t$ given the the sequence of observations $\langle \boldsymbol{o}\rangle_t = \boldsymbol{o}_t \ldots \boldsymbol{o}_1$.

The posterior pdf $p(\boldsymbol{q}_t|\langle \boldsymbol{o}\rangle_t)$ can be recursively calculated using Bayes' formula

$$p(\boldsymbol{q}_t|\langle \boldsymbol{o}\rangle_t) = \frac{1}{c} p(\boldsymbol{o}_t|\boldsymbol{q}_t) p(\boldsymbol{q}_t|\langle \boldsymbol{o}\rangle_{t-1}) \qquad (1)$$

with c being a normalizing constant. Equation (1) involves two pdfs, first of which is the so called temporal prior $p(\boldsymbol{q}_t|\langle \boldsymbol{o}\rangle_{t-1})$ that realizes the recursion and that is obtained by

$$p(\boldsymbol{q}_t|\langle \boldsymbol{o}\rangle_{t-1}) = \int p(\boldsymbol{q}_t|\boldsymbol{q}_{t-1}) p(\boldsymbol{q}_{t-1}|\langle \boldsymbol{o}\rangle_{t-1})\, d\boldsymbol{q}_t. \qquad (2)$$

The pdf $p(\boldsymbol{q}_t|\boldsymbol{q}_{t-1})$, the state transition, models the evolution of the object's state over time. It is assumed that the state depends only on its direct predecessor and therefore state evolution shows the Markovian property.

The second pdf needed in (1) is the likelihood $p(\boldsymbol{o}_t|\boldsymbol{q}_t)$. The likelihood is used to model the process that generates an observation given a certain state of the object. For example, it could model the perspective projection of the object onto the image plane of a camera by additionally taking into account background clutter and sensor noise. Evaluating the likelihood for an observation measures how likely this observation is, given a certain state. In the context of (1) the likelihood can be seen as weighting the predicted states from the temporal prior according to the actual available observation $\boldsymbol{o}_t$. In Section 3.2 we present the likelihood pdf that has been used later on for the experiments.

3.1 Particle Filters

The problem with equations (1) and (2) for sequential state estimation is, that for most cases, they can not be evaluated in closed form. In general not only the posterior pdf but also the likelihood pdf is multimodal and not analytically describable. These two properties finally make the integral in (2) intractable. Only in few special cases there exists a solution to the recursion and the posterior pdf can be computed. One of these special cases is handled by the Kalman filter and its relatives [1]. If all involved pdfs are Gaussian and if the state transition and the observation process can be described by linear equations superimposed by white Gaussian noise, then the posterior pdf remains Gaussian with the mean and covariance calculated by the Kalman filter equations.

This special case of state estimation is not satisfactory for applications in computer vision. For example, just the use of the simple pinhole camera model introduces a nonlinearity in the observation process. Additionally, the likelihood pdf is multimodal in general due to clutter, i.e., detected features not originating from the object but from the background. This all causes the posterior pdf to become multimodal and therefore prohibits the use of the Kalman filter.

Particle Filters (PF) – firstly introduced to the computer vision community under the name Condensation algorithm by [8] – offer a way out of this situation. In PFs the posterior pdf is approximated by a finite set of N weighted samples. Each sample represents a certain state and its weight is proportional to the value of the posterior for this state. The PF propagates these samples through time according to the following scheme. At first, in the so called *prediction step*, N new i.i.d. samples are drawn from the current posterior sample set. To each of these new samples the dynamic model is applied ($t \rightarrow t+1$). According to the Monte-Carlo method of composition [18] this leads to i.i.d. samples from the temporal prior (cf. (2)). Afterwards, in the *update step*, these samples get weighted with respect to the current observation and the likelihood. Now, a sample set representation of the posterior at time $t+1$ exists, and the filter restarts with the prediction step.

3.2 Likelihood Computation

We model the distribution of the observations given a certain state by a Gibbs distribution, which is a commonly used technique [3]. The pdf of the distribution is given by

$$p(\boldsymbol{o}_t|\boldsymbol{q}_t) = \frac{1}{z}\exp\left[-E(\boldsymbol{o}_t|\boldsymbol{q}_t)\right] \qquad (3)$$

with z being a normalizing constant. The term $E(\boldsymbol{o}_t|\boldsymbol{q}_t)$ (with $E(\boldsymbol{o}_t|\boldsymbol{q}_t) \geq 0$) can be interpreted as an energy. This energy should be high, if the image data $\boldsymbol{o}_t$ does not correspond well to the data that is predicted from the state $\boldsymbol{q}_t$. The energy should be low for a good match.

In our case, the light-field object model predicts an image of the object assuming a certain state. This image has to be compared to the actual observation. Therefore, the energy term $E(\boldsymbol{o}_t|\boldsymbol{q}_t)$ has to measure the similarity between the predicted observation and a certain target region in the actually observed image.

The energy $E(\boldsymbol{o}_t|\boldsymbol{q}_t)$ was defined by means of the *normalized correlation coefficient* (NCC). NCC is a common similarity metric between images used for template matching. Because the likelihood definition (3) requires high positive energy for bad matches, we used the reciprocal of the absolute NCC. Additionally, the absolute NCC is increased to the power of α with $\alpha > 1$. We call α a *strictness* parameter because the higher this parameter is chosen, the better the match has to be to finally achieve a high likelihood.

4 Experimental Results

We used three different objects for our experiments (cf. Figure 3): a stuffed elk (*elk*), a Santa Claus made of porcelain (*santa*), and a tape cassette (*tape*).

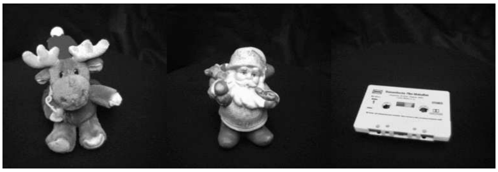

Figure 3: Objects used for experiments. From left to right: *elk*, *santa*, *tape*

For each of these objects a light-field object models has been constructed. Each object was recorded from 817 camera positions on a hemisphere with the turntable-arm assembly described in Section 2.

To evaluate accuracy of the proposed tracking technique ground truth data needs to be available. Therefore, we recorded the objects again from camera positions on the hemisphere that were different from those used for the generation of the light-field object models. We used 687 images for the *elk*, 507 images for the *santa*, and 508 images for the *tape*. The camera positions were stored with the images. Afterwards, object motion can be simulated by defining different paths through those camera positions. For our experiments we used three different paths of length 72 that were defined by sampling sinusoidal curves over the arm and turntable angles. The corresponding camera position to each sample has been obtained by using the respective nearest neighbor, therefore some camera positions may appear multiple times in one path. Figure 4 depicts the camera positions together with the paths.

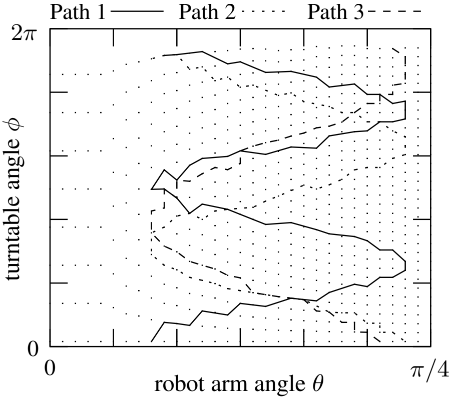

Figure 4: Three different paths of length 72 through the camera positions (dots) on the hemisphere for the object *elk*. For the other objects the paths are similar.

We deployed a particle filter for tracking the objects on their paths (i.e., actually, the camera positions are estimated). At each timestep the estimated state is taken to be the sample mean of the posterior density, hence a minimum mean square estimator (MMSE) is realized. For the experiments presented here, we restrict the state to only contain the 3-D pose parameters, namely, the two angles ϕ and θ, their velocity, and their acceleration, i.e.,

$$\boldsymbol{q}_t = \left(\phi, \dot{\phi}, \ddot{\phi}, \theta, \dot{\theta}, \ddot{\theta}\right)^{\mathrm{T}} \quad . \qquad (4)$$

For these experiments here scaling is not yet considered. The distance of the cameras on the hemisphere to the center of gravity of the objects is assumed to be fixed. Therefore, no position information is included in the state vector. The dynamic of the object is modeled by a linear equation whereby the state transition noise was estimated from the paths.

In advance to the tracking experiments we determined the best suited strictness parameter α (cf. Section 3.2). Each object was tracked on all three paths at an image resolution of 128^2 with $\alpha = 1, \ldots, 10$ using 100 samples for the particle filter. The estimation errors in θ and ϕ for each of the 90 runs were evaluated and averaged. To enable comparison of the results, the averaged estimation errors for a certain α were weighted by the number of successfully completed trackings with this α (e.g., for $\alpha = 1$ only 2 out of the 9 trackings succeeded, for $\alpha = 2$ it were 6). The best results were achieved for $\alpha = 8$, so we decided to use this value for all subsequent experiments.

In our main experiments we investigated how the performance of the tracking depends on the number of samples and on the chosen image resolution. Each object was tracked on each path using three different image resolutions: 64^2, 128^2, and 256^2, and three different numbers of samples for the particle filter: 50, 100, and 200, resulting in a total of 81 experiments.

For some of the combinations it occurred that the tracker lost the object and did not keep tracking it along the whole path. The numbers of successful trackings for different resolutions and numbers of samples are summarized in Table 1. The figures in the table support the expected: the higher the number of samples, the higher the number of successful trackings. Detailed evaluation has shown that the *elk* was the most difficult object to track.

	n out of 9	Resolution		
		64^2	128^2	256^2
Samples	50	6	5	5
	100	8	8	7
	200	9	9	8

Table 1: Number of successfully tracked paths over all objects for different resolutions and numbers of samples.

After having sorted out the experiments for which tracking failed, accuracy of the tracker can be evaluated. Accuracy is measured by calculating the mean absolute estimation error in the angles θ and ϕ against the known ground truth data of the constructed motion paths. The results of the evaluations are given in Table 2 for the angle θ and in Table 3 for the angle ϕ. From these tables it can be seen that the angle θ was estimated more accurately than the angle ϕ. The averaged mean absolute estimation error for θ is 2.522 degrees by a standard deviation of 0.085 degrees. This small variation in error suggests that the accuracy in estimating θ is nearly independent on the chosen number of samples and resolution. But, this is not the case for the angle ϕ. The averaged mean absolute estimation error is 3.281 degrees by a standard deviation of 0.36 degrees. The figures in Table 3 show that estimation becomes more accurate with an increasing number of samples, but it becomes more inaccurate with higher image resolution. This effect – of which the cause we have not yet fully understood – is very useful for the purpose of real-time object tracking where computation time is limited and in general a compromise between accuracy and speed needs to be implemented. In our case the processing of one sample, including the rendering of an image from the light-field, takes about 0.14 seconds at a resolution of 64^2, 0.19 seconds at 128^2, and 0.21 seconds at 256^2 on a Pentium III, 800 MHz. Therefore, the figures suggest to use 100 samples for the particle filter and an image resolution of 64^2 to achieve the desired compromise.

	Error in θ	Resolution			
		64^2	128^2	256^2	Mean
Samples	50	2.913	2.441	2.154	2.503
	100	2.543	2.707	2.767	2.672
	200	2.610	2.324	2.242	2.392
	Mean	2.688	2.491	2.388	

Table 2: Mean absolute estimation errors of turntable angle θ for different resolutions and numbers of samples, given in degrees averaged over all successfully tracked paths and objects.

The qualities of the tracking and the rendered images are demonstrated in Figure 5 exemplarily for the object *elk* on path 1 tracked with 100 samples at

Error in ϕ		Resolution			
		64^2	128^2	256^2	Mean
Samples	50	3.562	4.618	4.320	4.167
	100	3.144	3.173	3.323	3.213
	200	2.400	2.375	2.614	2.463
	Mean	3.035	3.388	3.419	

Table 3: Mean absolute estimation errors of robot arm angle ϕ for different resolutions and numbers of samples, given in degrees averaged over all successfully tracked paths and objects.

a resolution of 64^2. For each timestep we rendered an image from the light-field object model corresponding to the sample mean of the posterior distribution, i.e., corresponding to the estimated state. These images are placed beside the original images from the sequence to allow for visual comparison. It can be seen that the object's pose is estimated nearly accurate.

5 Conclusion

In this contribution we demonstrated how light-field object models could be used for 3-D object tracking with pose estimation. The main advantages of the light-field approach are, first, that model generation is simple compared to geometric object models, and second, that models could be generated for nearly every kind of object without any prior knowledge about the object's geometry. We showed, how these models fit easily into the framework of statistical object tracking by defining a likelihood function based on a Gibbs distribution. The presented experimental results circumstantiate that the proposed approach is suitable for the task of pose estimation.

In future experiments the effect of decreasing accuracy of the state estimator at increasing image resolution needs further investigations. Perhaps, the effect is due to some kind of noise reduction or blurring that arises from image scaling. Especially, it should be examined, if accuracy still increases with a further image size reduction.

Additional improvements could be applied to the light-field and the model generation process. It has been mentioned before in Section 2 that the camera positions need not be calibrated by hand but can be calibrated automatically by tracking point features resulting in point correspondences between the images. The benefits would be on the one hand a reduction of the amount of interaction required, and on the other hand the resulting availability of depth information from the calibration process. Nevertheless for calibration to work well on a wide enough variety of objects tracking and calibration algorithms still need to be improved. Apart from that, depth maps could also be generated by the application of stereo algorithms, which may even yield the better results.

Currently, we are working on experiments with tracking the objects in image sequences recorded in an office environment with varying illumination and background. To accomplish this task, we have to deal with problems that occur in this general situation, e.g., object scaling, automatic initialization of the tracking process, calibration of the camera and registration of the calibration parameters to the light-field parameters.

References

[1] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association*. Academic Press, Boston, San Diego, New York, 1988.

[2] Michael J. Black and Allan D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In B. Buxton and R. Cipolla, editors, *Computer Vision – ECCV'96*, volume 1, pages 329–342, Cambridge, 1996.

[3] A. Blake and A. Yuille, editors. *Active Vision*. MIT Press, Cambridge, Massachusetts, London, England, 1992.

[4] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proceedings SIGGRAPH '96*, pages 43–54, New Orleans, August 1996. ACM Press.

[5] B. Heigl, J. Denzler, and H. Niemann. On the application of lightfield reconstruction for statistical object recognition. In *European Signal Processing Conference (EUSIPCO)*, pages 1101–1104, 1998.

[6] B. Heigl, J. Denzler, and H. Niemann. Combining computer graphics and computer vision for probabilistic visual robot navigation. In J.G. Verly, editor, *Enhanced and Synthetic Vision 2000*, volume 4023 of *Proceedings of SPIE*, pages 226–235, April 2000.

[7] B. Heigl, R. Koch, M. Pollefeys, J. Denzler, and L. Van Gool. Plenoptic Modeling and Rendering from Image Sequences Taken by a Hand–Held Camera. In *Mustererkennung 1999*, pages 94–101, Heidelberg, 1999. Springer–Verlag.

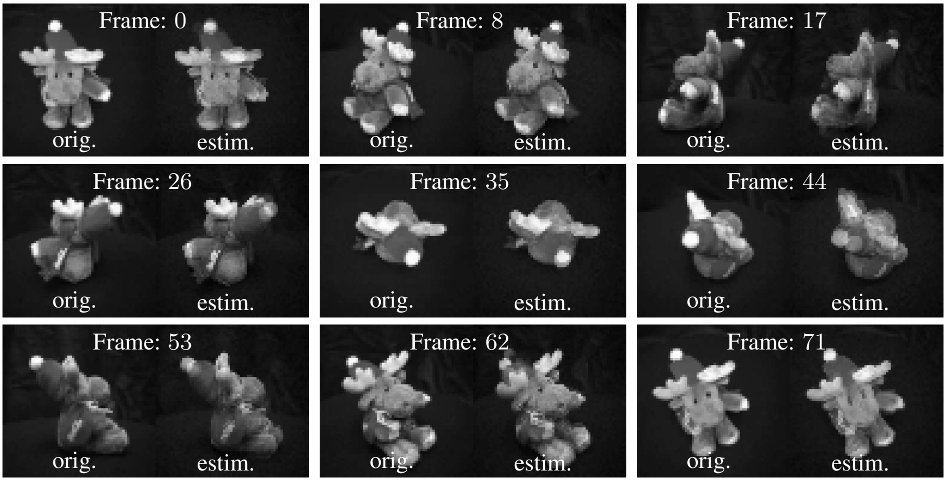

Figure 5: Comparison between the original image (left) of the test sequence for object *elk* on path 1 tracked with 100 samples at a resolution of 64^2 and the rendered image (right) from the light-field according to the estimated state. Every ninth image from the sequence is shown from left to right and top to bottom.

[8] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In B. Buxton and R. Cipolla, editors, *Computer Vision – ECCV'96*, volume 1, pages 343–356, Cambridge, 1996.

[9] R. Koch, M. Pollefeys, B. Heigl, L. van Gool, and H. Niemann. Calibration of hand-held camera sequences for plenoptic modeling. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 1, pages 585–591, Corfu, Greece, September 1999. IEEE Computer Society Press.

[10] D. Koller, K. Daniilidis, T. Thorhallson, and H. Nagel. Model–based object tracking in traffic scenes. In G. Sandini, editor, *Computer Vision - ECCV 92*, pages 437–452, Berlin, Heidelberg, New York, London, 1992. Lecture Notes in Computer Science.

[11] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings SIGGRAPH '96*, pages 31–42, New Orleans, August 1996. ACM Press.

[12] E. Marchand, P. Bouthemy, F. Chaumette, and V. Moreau. Robust real–time visual tracking using a 2d–3d model–based approach. In *Proceedings of the Seventh International Conference on Computer Vision*, pages 262–268, Kerkyra, Greece, 1999.

[13] H. Murase and S.K. Nayar. Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, January 1995.

[14] Kevin Nickels and Seth Hutchinson. Integrated object models for robust visual tracking. In *Proceedings of Workshop on Robust Vision for Vision-Based Controlof Motion, Leuven, Belgium*, 1998.

[15] H. Schirmacher, C. Vogelgsang, H.-P. Seidel, and G. Greiner. Efficient free form light field rendering. In *Proceedings of Vision, Modeling, and Visualization 2001*, pages 249–256, Stuttgart, Germany, November 2001.

[16] H. Sidenbladh, F. De la Torre, and M.J. Black. A framework for modeling the appearance of 3d articulated figures. In *Proceeding of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, France*, pages 368–375, Los Alamitos, California, 2000. IEEE Computer Society.

[17] Katrin Stark. A method for tracking the pose of known 3-d objects based on an active contour model. Report, Technical University Dresden, Computer Science Departement, 01062 Dresden, Germany, 1996.

[18] M.A. Tanner. *Tools for Statistical Inference*. Springer Verlag, London, Berlin, Heidelberg, New York, Paris, Tokyo, Hong Kong, Budapest, 1993.