Plenoptic Modeling of 3D Scenes with a Sensor-augmented Multi-Camera Rig

Reinhard Koch, Jan-Michael Frahm, Jan-Friso Evers-Senne, Jan Woetzel

Institute of Computer Science and Applied Mathematics, Multimedia Information Processing Group
Christian-Albrechts-University of Kiel, Olshausenstr. 40, 24098 Kiel, Germany

Email: *rk@informatik.uni-kiel.de*, Web: *www.mip.informatik.uni-kiel.de*

ABSTRACT

We propose a system for robust modeling and visualisation of complex outdoor scenes from multi-camera image sequences and additional sensor information. A camera rig with one or more fire-wire cameras is used in conjunction with a 3-axis rotation sensor to robustly obtain a calibration of the scene with an uncalibrated structure from motion approach. Dense depth maps are estimated with multi-viewpoint stereo and the scene is visualized from a plenoptic representation of the scene.

1 INTRODUCTION

This contribution discusses 3D scene reconstruction and visualization from real image streams that are recorded with a portable, freely moving hand-held camera rig. This approach allows to acquire 3D scene models from complex real-world scenes for visual rendering purposes.

We focus our attention on the acquisition of long image sequences in uncontrolled outdoor environments. Here we face many problems, since neither the geometric scene structure nor the scene surface properties or the lighting conditions are known or controllable. In addition, there may be objects moving in the scene, be it with stochastic or repetitive motion like swaying tree branches, or systematic motion like moving people or cars. Finally, when operating the camera by hand, the motion of the camera trajectory is often jerky, with small bouncing moves or sudden rotations. Therefore, the camera tracking and calibration must operate very robustly and be tolerant to such problems.

One approach to camera calibration and dense scene reconstruction is Structure from Motion (SfM) [19]. SfM tries to model the 3D scene and the camera motion geometrically and captures scene details on polygonal (triangular) surface meshes. A limited set of camera views of the scene is sufficient to reconstruct the 3D scene. Texture mapping adds the necessary fidelity for photo-realistic rendering of the object surface. In this approach, dense and accurate 3D depth estimates are needed for realistic image rendering from the textured 3D surface model. Deviation from the true 3D surface will distort the rendered images.

Another approach to the problem is plenoptic modeling [17]. Lightfield rendering [16] and the lumigraph [7] approach have received a lot of attention, since they can capture the appearance of a 3D scene from images only, without the explicit use of 3D geometry. Thus one may be able to capture objects with very complex geometry that can not be modeled otherwise. Basically one caches views from many different directions all around the scene and interpolate new views from this large image collection. For realistic rendering, however, very many views are needed to avoid interpolation errors for in-between views.

The problem common to both approaches is the need to calibrate the image sequence. Recently it was proposed to combine a structure from motion approach with plenoptic modeling to generate lightfields from uncalibrated hand-held camera sequences [13]. When generating lightfields from a hand-held camera sequence, one typically generates images with a specific distribution of the camera viewpoints. Since we want to capture the appearance of the object from all sides, we will sample the viewing sphere, thus generating a *mesh of view points*. To fully exploit hand-held sequences, we will therefore have to deviate from the regular lightfield data structure and adopt a more flexible rendering data structure based on the viewpoint mesh. Another important point in combining SFM and lightfield rendering is the use of scene geometry for image interpolation. The geometric reconstruction yields a geometric approximation of the real scene structure that might be insufficient when static texture mapping is used. However, view-dependent texture mapping as in [3] will adapt the texture dynamically to a static, approximate 3D geometry.

In the following sections we will discuss our approach to model the type of scenes as described above. We will first give some background on the principal Structure from Motion approach that is used to calibrate the cameras and to obtain 3D scene structure. Then we will discuss the problems of the approach and how to overcome some of the difficulties. Finally we will give first results of the approach with respect to scene rendering.

2 CAMERA CALIBRATION AND 3D SCENE RECONSTRUCTION

Uncalibrated *Structure from Motion* is used to recover camera calibration and scene geometry from images of a static scene alone without the need for further scene or camera information. Faugeras and Hartley first demonstrated how to obtain uncalibrated projective reconstructions from image point matches alone [5, 8]. Beardsley et al. [2] proposed a scheme to obtain projective calibra-

tion and 3D structure by robustly tracking salient feature points throughout an image sequence. This sparse object representation outlines the object shape, but does not give sufficient surface detail for visual reconstruction. Highly realistic 3D surface models need a dense depth reconstruction and can not rely on few feature points alone.

In [19] the method of Beardsley was extended in two directions. On the one hand the projective reconstruction was updated to metric even for varying internal camera parameters, on the other hand a dense stereo matching technique [4] was applied between two selected images of the sequence to obtain a dense depth map for a single viewpoint. From this depth map a triangular surface wire-frame was constructed and texture mapping from one image was applied to obtain realistic surface models. In [12] the approach was further extended to multi-viewpoint depth analysis. The approach can be summarized in 3 steps:

- Camera self-calibration and metric structure is obtained by robust tracking of salient feature points over the image sequence,
- dense correspondence maps are computed between adjacent image pairs of the sequence,
- all correspondence maps are linked together by multiple view point linking to fuse depth measurements over the sequence.
- textured triangular surface models are generated for geometric scene reconstruction and for scene rendering.

2.1 Calibration of a mesh of viewpoints

When very long image sequences have to be processed with the approach described above, there is a risk of calibration failure due to several factors. For one, the calibration is built sequentially by adding one view at a time. This may result in accumulation errors that introduce a bias to the calibration. Secondly, if a single image in the sequence is not matched, the complete calibration fails. Finally, sequential calibration does not exploit the specific image acquisition structure used in this approach to sample the viewing sphere.

In [13] a multi-viewpoint calibration algorithm has been described that allows to actually weave the viewpoint sequence into a connected viewpoint mesh. Starting from image pairs, corresponding point matches are computed and the Fundamental Matrix F can be estimated. F maps a point in one image of an image pair to its corresponding epipolar line in the other image. From F one can derive a projective camera and instantiate a first set of 3D feature points by triangulating the corresponding image points. The 3D feature points are determined such that their reprojection error in the images is minimized. The 3D object points serve as the *memory* for consistent camera tracking, and it is desirable to track the projection of the 3D points through as many images as possible. This process is repeated by adding new viewpoints and correspondences from the recorded image sequence. Finally constraints are applied to the cameras to obtain a metric reconstruction. A detailed account of this approach can be found in [18, 19].

Since we are collecting a large amount of images from all possible viewpoints distributed over the viewing sphere, it is no longer reasonable to consider a sequential processing along the sequence frame index alone. Instead we consider all images of the sequence simultaneously and compute matches between them in order to robustly establish image relationships between all nearby images. The simultaneous matching creates a connected mesh of all nearby camera viewpoints that exploits the actual 2D topology between the views rather than following the one-dimensional camera path (see [13, 14]).

2.2 Depth map estimation

Once we have retrieved the metric calibration of the cameras we can use image correspondence techniques to estimate scene depth. We rely on stereo matching techniques that were developed for dense and reliable matching between adjacent views. The small baseline paradigm suffices here since we use a rather dense sampling of viewpoints, as si the case when recording sequences with video frame rate.

For dense correspondence matching an area-based disparity estimator is employed. The matcher searches at each pixel in one image for maximum normalized cross correlation in the other image by shifting a small measurement window (kernel size 7x7) along the corresponding epipolar line. Dynamic programming is used to evaluate extended image neighborhood relationships and a pyramidal estimation scheme allows to reliably deal with very large disparity ranges [4]. The geometry of the viewpoint mesh is especially suited for further improvement with a multi viewpoint refinement [12]. Each viewpoint is matched with all adjacent viewpoints and all corresponding matches are linked together to form a reliable depth estimate. Since the different views are rather similar we will observe every object point in many nearby images. This redundancy is exploited to improve the depth estimation for each object point, and to refine the depth values to high accuracy. This approach could be exchanged for novel multi-stereo-approaches like Graph Cut approaches [15] or the MaxFlow algorithm [20]. For a recent review on stereo algorithms see also [21].

2.3 View-dependent plenoptic rendering from the modeled scene

After camera calibration and stereoscopic depth estimation, we now have a set of calibrated depth maps and associated color images at hand. These data could be utilized to reconstruct a 3D surface model of the 3D scene. However, due to the possibly very complex scene geometry with occlusions and because of varying surface reflectance, we are normally not able to create high fidelity surface models. It may also not be possible to reconstruct a globally consistent model due to systematic errors in calibration. Therefore we have to relax the very strict requirement of obtaining a *globally consistent* model, but we can still hope to recover a *locally consistent* model. A model is locally consistent if we can compute an approximate model from the depth maps of few nearby cameras

and render views of the scene that look similar to the original views from nearby view points. In fact, rendering from a locally consistent model is equivalent to image-based rendering with depth-compensated image interpolation. We select the nearest original views and according depth maps to render novel views by image warping. This approach is called plenoptic rendering since we recover novel views by rendering from the plenoptic function of all possible view points of the scene.

One possible method for plenoptic rendering is *light-field rendering*[16]. To create a lighfield model for real scenes, a large number of views from many different angles are taken. Each view can be considered as a bundle of light rays passing through the optical center of the camera. The set of all views contains a discrete sampling of light rays with according color values and hence we get discrete samples of the plenoptic function. The light rays which are not represented have to be interpolated.

To render a new view we suppose to have a virtual camera pointing towards the scene. For each pixel we can determine the position of the corresponding virtual viewing ray. The nearer a recorded ray is to this virtual ray the greater is its support to its color value. So the general task of rendering views from a collection of images will be to determine those viewing rays which are *nearest* to the virtual one and to interpolate between them depending on their proximity.

Direct linear interpolation between the viewpoints introduces a blurred image with ghosting artifacts. In reality we will always have to choose between high density of stored viewing rays with high data volume and high fidelity, or low density with poor image quality. If we know an approximation of the scene geometry from local depth maps, the rendering result can be improved by an appropriate depth-dependent warping of the nearest viewing rays as described in [7].

Having a sequence of images taken with a hand–held camera, in general the camera positions are not placed at the grid points of the viewpoint plane. We have solved the problems described in the last section by relaxing the restrictions imposed by the regular lightfield structure, and we will render views directly from the calibrated sequence of recorded images using local depth maps. Without loosing performance we can directly map the original images onto a surface viewed by a virtual camera with projective texture mapping that is supported by graphics hardware.

Following the lightfield approach, we have to interpolate between neighboring views to construct a specific virtual view. Considering the fact mentioned above that the *nearest* rays give the best support to the color value of a given ray, we conclude that those views give the most support to the color value of a particular pixel whose projection center is closest to the viewing ray of this pixel. This is equivalent to the fact that those real views give the most support to a specified pixel of the virtual view whose projected camera centers are close to its image coordinate. We restrict the support to the nearest three cameras (see figure 1). To determine these three neighbors we project all camera centers into the virtual image and perform a 2–D triangulation. Then the neighboring cameras of a pixel are determined by the corners of the triangle which this pixel belongs to. The texture of such a triangle — and consequently a part of the reconstructed image — is drawn as a weighted sum of three textured triangles.

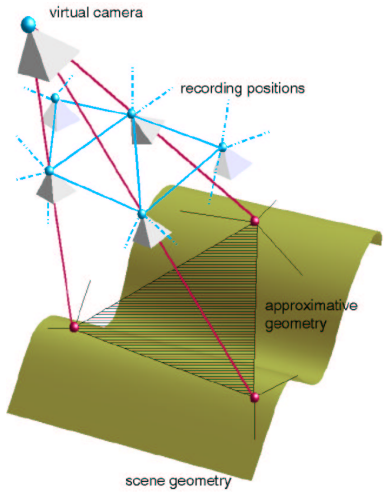

Figure 1: Drawing triangles of neighboring projected camera centers and approximating scene geometry by one plane for the whole scene or for one camera triple.

The results can be further improved by considering local depth maps. Spending more time for each view, we can calculate the approximating plane of geometry for each triangle in dependence on the actual view. As the approximation is not done for the whole scene but just for that part of the image which is seen through the actual triangle, we don't need a consistent 3-D model but we can use the local depth maps. For more details we refer to [10].

3 Improving robustness for complex scenes

The approach as outlined above is highly dependent on the input scene, since the calibration and camera tracking relies on observable image features. Because the environment of complex outdoor sequences can not be controlled, all possible parameter changes like changing light, directional surface reflectance, moving objects, and the lack or abundance of surface texture may influence the estimation, causing in general the algorithms to be not very robust. Also, for general hand-held cameras, there may be a lot of unsteady movement, caused by the bouncing of the hand and by rotation of the camera. We have investigated different modalities to increase the robustness of the calibration with respect to these modalities. Currently we have focused on two approaches, namely using stereoscopic or multi-camera rigs, and on the use of additional rotational information, acquired by a 3-axis rotation sensor.

3.1 Structure and motion estimation with a moving multi-camera rig

We employ a rig with two or more digital fire-wire cameras that are mounted onto a pole. The position of the cameras on the pole can be changed freely according to the requirements of the scene. Typically we mount

Figure 2: Left: Original images 1, 25, 50, 75, 100 of upper camera of stereo rig. Right: Depth maps for the images to the left (dark=near, light=far, black=undefined).

the cameras one above each other in 30-50 cm distance, looking towards the scene. We do not need to calibrate the rig beforehand since all calibration will be done automatically from the sequence. We can also change the relative camera positions during the shooting if needed.

The advantage of using a multi-camera rig is twofold. For one we have, at each time instant, two or more images of the scene taken from different view-points, which by definition is now static (not-moving). Hence one important prerequesite of the SfM approach is fulfilled. This holds even if we change the relative pose of the cameras on the pole during image acquisition. We also can guarantee that there is always sufficient base line distance between the cameras in order to reconstruct local depth information. Secondly, we automatically generate a 2D viewpoint mesh of the scene if we move the camera rig horizontally through the scene, while the different cameras are mounted vertically above each other. Each camera generates a horizontal track in time, while for each time instant a rigid vertical track is formed by the cameras. This construction allows very robust estimation of the viewpoint mesh if at least two cameras are tracked.

3.1.1 Scene modeling with moving stereo rig

To test the approach, we acquired a stereoscopic image sequence of a parking lot. The scene itself contained trees, bushes, street and parking lot with some cars at various distances, and buildings in the background. The images were taken by two digital FireWire cameras of size 1024x768 with 45 degree opening angle, that were grabbed directly to a laptop with about 3 fps. The cameras were mounted vertically on a pole about 35 cm apart and moved horizontally sideways by hand for about 30 m. Each camera took 100 images, each about 30 cm apart horizontally. The depth extend of the scene was from 2 m to about 80 m for the background. Some example images for the scene are shown in figure 2.

The calibration was able to track the camera sequence with high accuracy. About 9000 3D scene feature points were generated from the 200 images. Each 3D point was visible in 8.2 images at average, with an average reprojection error of 0.23 pixel. The estimated scaled camera trajectory was accurate to about 1% positional error. Figure 3 shows different views of the 3D feature points and the camera trajectory.

From the cameras, dense depth maps were computed with an average fillrate of 76% and an average relative depth error of 0.45% in the range of 1 m to 80 m distance. The resulting scaled depth maps are shown in figure 2.

So far we could obtain only preliminary rendering results in figure 4. The complete composite scene was rendered by the superposition of 5 partial reconstructions of 5 depth map, and a closeup view shows a geometric reconstruction of the scene from a middle view. So far, no plenoptic rendering was performed.

3.2 Exploiting additional orientation data

The system described above is already very robust to scene feature changes. Yet there is another problem related to the image acquisition. The SfM approach needs initial image feature correspondences to compute the F-Matrix. Because initially we do not know the epipolar constraint, we must allow possible correspondence matches over a large search range in the image. If the rig would be moving with a pure translation only, then the search range would be affected by the displacement of the camera center and the scene depth only. This limits the possible search range since the camera motion between two frames is typically small. If, on the other hand, the camera is rotated arbitrarily by walking or shaking of the hand, a very large displacement of corresponding image features is possible. This displacement is caused by a 2D homograpy and no relevant depth information is contained in it. When the rotation and the intrinsic calibration is known then we can undo the rotation by the proper inverse homograpy and stabilize the image pair. To further improve the performance of the above mentioned techniques we are investigating to use rotation information from an external sensor.

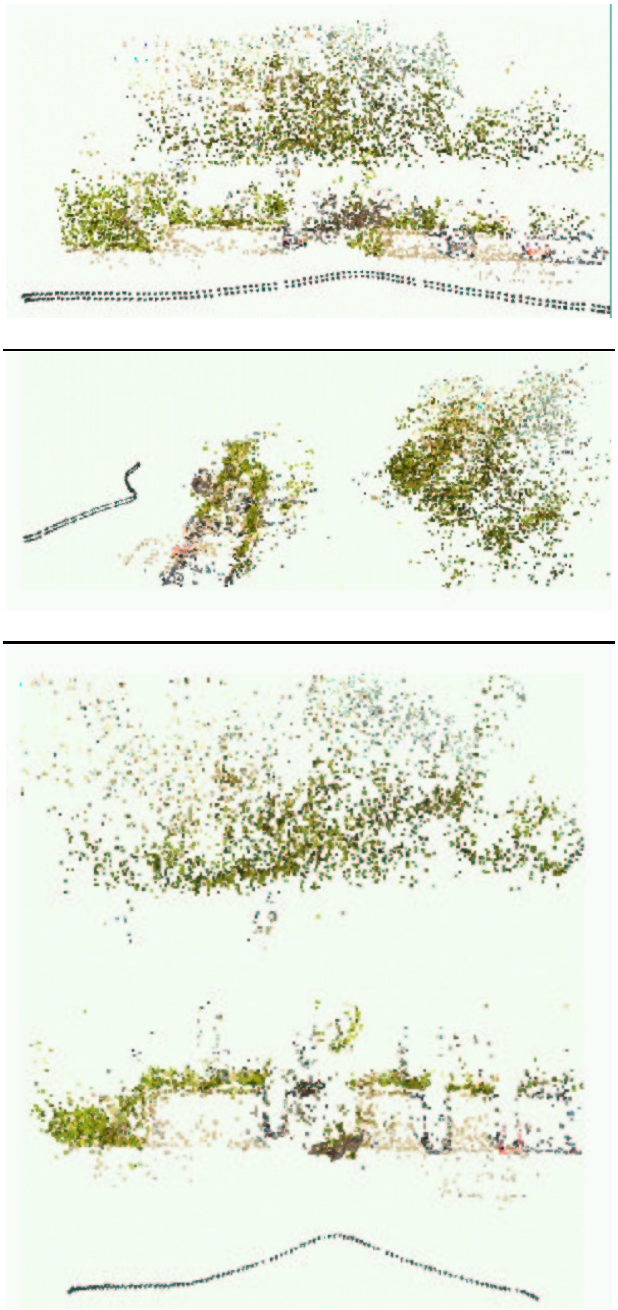

Figure 3: Perspective views of 3D feature points. One may note the feature positions where cars, trees and bushes are. Each camera position is depicted as little pyramid. Top: Front view of tracked 3D feature points (colored with scene color) and the stereo camera trajectory (similar view to Fig. 4, Top). Middle: Side view of scene. Bottom: Top view of scene points and camera tracks.

Figure 4: Rendered views of the scene: Top and Middle: Novel view rendered from a scene composed from 5 view points by direct mapping from raw depth maps. White spots indicate missing image data where no depth was available. Bottom: Closeup view from 3D surface model with depth interpolation.

3.2.1 *Sensor fusion of orientation sensor and image data*

One practical issue arising from orientation sensor fusion is the synchronization of sensor readings with the camera shutter time. The sensor data and camera acquisition need to be aligned in time. To align cameras and rotation sensors we use a precalibration step with the camera rotating about its optical center. In this case we will align the rotation sensor with the camera by using the homography H_j from the camera rotation between time j and $j+1$. We can then compare the rotation information of the sensor with the estimated rotation of the camera and align both in time such that the rotation signature has maximum correlation.

For a rotating camera with fixed internal parameters and therefore fixed K we no not need to calibrate the internal camera parameters to compute the camera rotation by

$$R_i = K^{-1} H_j K. \tag{1}$$

In this case the homography H_j between two views is for constant K a conjugated[1] rotation matrix R_i if R_i is the rotation matrix which fulfills (1). It is known [9] that the

[1] Matrices A and B are conjugated if $A = CBC^{-1}$ for some matrix C.

rotation matrix R_i and homography H_j have the same eigenvalues

$$eig(R_i) = eig(H_j).$$

since a conjugated matrix $A = C^{-1}BC$ has the same eigenvectors as the matrix B.The estimated homography H_j between views j and $j+1$ is only determined up to scale s_j. Therefore we have to deal with the eigenvalues of a scaled matrix $s_j H_j$. The eigenvalues of a scaled matrix are also scaled by s_j[9]

$$eig(s_j \cdot H_j) = s_j \cdot eig(H_j) \quad \text{with } s_j \in \mathbb{R}.$$

For this case in [11] a similarity measure for homographies H^π over arbitrary planes π is associated with the vectors containing the eigenvalues in descending absolute value order (eigenvalue vectors)

$$sim_{eig}(R_i, H_j) = \frac{eig(R_i)^T eig(H_j)}{\|eig(R_i)\|\|eig(H_j)\|}, \qquad (2)$$

where $\|\cdot\|$ is a vector norm. It measures the parallelism of the eigenvalue vectors of rotation matrix R_i and homography H_j. For real valued eigenvalues, Eq. (2) provides the cosine of the angle between the two vectors. For homograhies H^∞ which are mapping via the plane at infinity the eigenvalues are not real valued. The eigenvalues of a rotation matrix R_i or of the resulting homography H^∞ are complex and a permutation of

$$eig(R_i) = [1, \cos\Phi_i + i\sin\Phi_i, \cos\Phi_i - i\sin\Phi_i]^T \quad (3)$$

and have absolute value 1 [6]. The eigenvector corresponding to eigenvalue 1 is the rotation axis for the rotion with angle Φ_i in 3D space. One can observe that the eigenvalue vector is a function of Φ_i

$$E(\Phi_i) = [1, \cos\Phi_i + i\sin\Phi_i, \cos\Phi_i - i\sin\Phi_i]^T. \quad (4)$$

This function has the same problems for inversion as the cosine has. Further the unknown scale s_j of the homography H_j^∞ is the absolute value of each component of the eigenvector. For this reason we devise a different matching criterion from eigenvalue vector. At first we have to eliminate the unknown scale s_j from the eigenvalues. The scale s_j is given by the norm of an arbitrary component of the eigenvalue vector $E(\Phi_i)$

$$\tilde{E}(\Phi_i) = \frac{1}{|E_{1,1}(\Phi_i)|}[1, \underbrace{\cos\Phi_i + i\sin\Phi_i}_{term_1}, \underbrace{\cos\Phi_i - i\sin\Phi_i}_{term_2}]^T. \qquad (5)$$

After normalization we compute the angle Φ_{mean} form the normalized eigenvalue vector $\tilde{E}(\Phi_i)$

$$\Phi_{mean} = \frac{1}{4} \sum_{term_1, term_2} \left(\arccos(\cos\Phi_i) + \arcsin(\sin\Phi_i)\right). \qquad (6)$$

Then we measure the distance between the rotation angle Φ^R_{mean} from R_i and the rotation angle Φ^H_{mean} from H_j^∞ by

$$\Delta_{\cos} = |\Phi^R_{mean} - \Phi^H_{mean}|. \qquad (7)$$

This similarity measure is equivalent to (2) but in the case of noise it is more robust because of the averaged angle.

Eq. (7) measures the similarity of the rotations given by a rotation matrix and a homography matrix which maps over the plane at infinity. To estimate the time-shift Δt between the tracker and the camera we estimate the camera homographies H_j and we calculate the rotation of the sensor. Then we evaluate the criterion (7) for a given time-shift for each camera and sum up the errors. This leads to a similarity measurement

$$sim_{\cos} = \sum_{cameras} \Delta_{\cos} \qquad (8)$$

between a sequence of orientation information and a camera sequence. The minimum of this criterion is the searched time-shift Δt.

The criterion has the problem that the same rotation angle about different rotation axis is a minimum of criterion, too. The singularities of the similarity measures (2) and (8) usually do not disturb the matching process because the sequence of different rotations compensates the singularities.

3.2.2 Experimental results for alignment

In this section the above mentioned techniques will be evaluated with real data. We use a tracker and three externally synchronized cameras mounted on a pole. The tracker delivers updated rotation data every 8 ms and the cameras are externally synchronized. Therefore all cameras have the same delay in time to the rotation sensor. Furthermore the tracker data is received later than the synchronization signal is sent to the cameras. In fig. 5 the estimated delay in time of each camera to the orientation sensor information is marked by a vertical line. Best rotation alignment between the tracker and all three cameras was obtained for a tracker delay between 62 and 78 ms, which is within the time resolution of the rotation tracker.

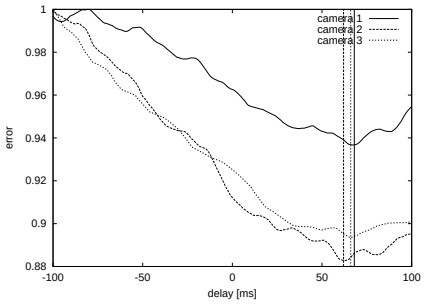

Figure 5: Measured error (7) for time delay for three synchronized cameras to orientation sensor. The estimated time delay Δt from criterion (8)is marked by a vertical line.

With the delay at hand, we can synchronize the tracker data with the camera acquisition and stabilize the image sequence w.r.t. the rotational component. Further tests have to show the impact of the stabilization onto the calibration results.

4 CONCLUSIONS

We have presented an approach to model complex outdoor scenes from images of an uncalibrated, freely moving camera rig. The underlying approach is uncalibrated structure from motion for camera tracking, dense multi-view stereo for depth map reconstruction, and plenoptic rendering for novel view synthesis. Our focus in this paper has been to improve the robustness by means of a stereo camera and an additional rotation sensor. Since we describe work in progress, not all modalities could be integrated, and not all advantages of the system could be explored. The use of a binocular stereo head has improved camera tracking stability very much and at the same time will allow plenoptic rendering for in-between views. So far, the rigid transformation between the cameras of the stereo head has not been exploited and we expect even better performance here. The extension of the rig to wider baseline between the views and to a multi-camera system is our current focus. The rotation has been synchronized with the image acquisition, and the rotation data will be integrated to reduce the feature search range and to facilitate the calibration. Further investigations will exploit the space-time sequence to allow also moving objects in the scene. Another important but open issue is the rendering of novel views of the scene from all available data, even in the presence of strong occlusions.

Acknowledgments

This research is in parts supported by the IST-Project ORIGAMI of the European Community.

REFERENCES

[1] L. de Agapito, E. Hayman, and I. Reid. Self-calibration of a rotating camera with varying intrinsic parameters. In *British Machine Vision Conference*, September 1998.

[2] P. Beardsley, P. Torr and A. Zisserman: 3D Model Acquisition from Extended Image Sequences. *ECCV 96*, LNCS 1064, vol.2, pp.683-695.Springer 1996.

[3] P. Debevec, Y. Yu, G. Borshukov: Efficient View-Dependent Image-Based Rendering with Projective Texture Mapping. Proceedings SIGGRAPH '98, ACM Press, New York, 1998.

[4] L.Falkenhagen: Hierarchical Block-Based Disparity Estimation Considering Neighborhood Constraints. Intern. Workshop on SNHC and 3D Imaging, Rhodes, Greece, Sept. 1997.

[5] O. Faugeras: What can be seen in three dimensions with an uncalibrated stereo rig. *Proc. ECCV'92*, pp.563-578.

[6] R. Franklin. Efficient rotation of an object. *IEEE Transactions on Computing*, November 1983.

[7] S. Gortler, R. Grzeszczuk, R. Szeliski, M. F. Cohen: The Lumigraph. Proceedings SIGGRAPH '96, pp 43–54, ACM Press, New York, 1996.

[8] R. Hartley: Estimation of relative camera positions for uncalibrated cameras. *ECCV'92*, pp.579-587.

[9] C. E. Pearson, editor. *Handbook of Applied Mathematics*, page 898. Van Nostrand Reinhold Company, New York, second edition, 1983.

[10] B. Heigl, R. Koch, M. Pollefeys, J. Denzler: Plenoptic Modeling and Rendering from Image Sequences taken by a Hand-Held Camera. Proceedings DAGM'99, Bonn, Sept. 1999.

[11] Yaron Caspi and Michal Irani. Alignment of non-overlapping sequences. In *ICCV*, 2001.

[12] R. Koch, M. Pollefeys, and L. Van Gool: Multi Viewpoint Stereo from Uncalibrated Video Sequences. *Proc. ECCV'98*, Freiburg, June 1998.

[13] R. Koch, M. Pollefeys, B. Heigl, L. Van Gool, H. Niemann: Calibration of Hand-held Camera Sequences for Plenoptic Modeling. Proc. of ICCV'99, Korfu, Greece, Sept. 1999.

[14] R. Koch, M. Pollefeys and L. Van Gool: Robust Calibration and 3D Geometric Modeling from Large Collections of Uncalibrated Images. Proceedings DAGM'99, Bonn, Sept. 1999.

[15] V. Kolmogorov and R. Zabih: Multi-Camera Reconstruction via Graph Cuts. In: Proceedings ECCV02, Springer LNCS 2352, pp 82-96, Heidelberg, 2002.

[16] M. Levoy, P. Hanrahan: Lightfield Rendering. Proceedings SIGGRAPH '96, pp 31–42, ACM Press, New York, 1996.

[17] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system", *Proc. SIGGRAPH'95*, pp. 39-46, 1995.

[18] M. Pollefeys, R. Koch, M. Vergauwen and L. Van Gool: Metric 3D Surface Reconstruction from Uncalibrated Image Sequences. In: 3D Structure from Multiple Images of Large Scale Environments. LNCS Series Vol. 1506, pp. 139-154. Springer-Verlag, 1998.

[19] M. Pollefeys, R. Koch and L. Van Gool: Self-Calibration and Metric Reconstruction Inspite of Varying and Unknown Intrinsic Camera Parameters. Int. Journal of Computer Vision 32(1), 7-27 (1999), Kluver 1999.

[20] S. Roy, M. Drouin: A maximum-flow formulation fo the n-camera stereo correspondence problem. In: Proc. ICCV pp. 492-499, Bombay, India, 1998.

[21] D. Scharstein, R. Szeliski: A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. Int. Journal of Computer Vision 2002.

[22] P.H.S. Torr: Motion Segmentation and Outlier Detection. PhD thesis, University of Oxford, UK, 1995.