MVA2005 IAPR Conference on Machine VIsion Applications, May 16-18, 2005 Tsukuba Science City, Japan

5-1

Dynamic Light Field Reconstruction and Rendering for Multiple Moving Objects

Ingo Scholz[1,*], Christian Vogelgsang[2,*], Joachim Denzler[3] and Heinrich Niemann[1]

[1]Universität Erlangen–Nürnberg
Lehrstuhl für Mustererkennung
Martensstr. 3, 91058 Erlangen
Germany

[2]Universität Erlangen–Nürnberg
Lehrstuhl für Graphische Datenverarbeitung
Am Weichselgarten 9, 91058 Erlangen
Germany

[3]Universität Jena
Lehrstuhl für Digitale Bildverarbeitung
Ernst-Abbe-Platz 2, 07743 Jena
Germany

email: scholz@informatik.uni-erlangen.de

Abstract

Light field reconstruction is still mostly limited to static scenes or is only applicable to dynamic scenes using sophisticated and costly hardware. In contrast to that, our contribution describes a system which allows the reconstruction of a light field of a scene including one or more rigidly moving objects using only one hand-held camera. By separating automatically tracked feature points into different objects, structure-from-motion algorithms can be applied for each object. The extension to long image sequences is done iteratively and the dynamic light field is created by merging the individual reconstructions and quantizing the object poses into distinct time steps.

For rendering purposes, an extension of the Unstructured Lumigraph is introduced which uses confidence maps to mark scene background, visible and invisible object poses.

1 Introduction

The light field [8] and the lumigraph [4], respectively, are by now well-established techniques for image-based rendering [1, 5]. They are especially well suited for reproducing images of real scenes or objects. For this purpose, the light field model usually consists of a collection of images or an image sequence of the scene from different viewing angles, along with the intrinsic and extrinsic camera parameters for each image.

Very often image data is acquired by mounting a calibrated camera on a gantry or robot arm which is moved around the object. However, using a hand-held camera for recording the required image sequences is cheaper and more flexible, although the camera pose information will not be available as easily. Structure-from-motion techniques, such as factorization methods, and camera calibration have to be applied to obtain the camera parameters [5].

So far, the light field was mostly restricted to the reproduction of static scenes. Allowing movement or deformation of objects in the scene adds a lot of complexity to the tasks of acquiring, storing and rendering images from the light field. Light fields which are variable in time are often referred to as *dynamic* light fields.

*This work was funded by the German Research Foundation (DFG) under grant SFB 603/TP C2. Only the authors are responsible for the content.

In this contribution we address the problem of dynamic light field acquisition considering only one hand-held camera, one or more rigid but permanently moving objects in the scene, and long image sequences of more than 100 frames. In addition to that, we propose a new rendering algorithm for the resulting light field which allows using all input images simultaneously for rendering each time step.

Object segmentation, camera pose and scene reconstruction are done using a multibody segmentation algorithm by Kanatani [6, 7] and a factorization [9] for each object. Since these algorithms are only applicable for rather short image sequences, we incorporate the multibody segmentation into the method proposed by Heigl [5], which extends an initial reconstruction of a short subsequence to long image sequences.

In order to create a complete dynamic light field model the independent reconstructions for each object are registered with each other and "time" steps of object motion are identified by a vector quantization of the relative camera positions. Different time steps of the final light field are rendered by creating mask matrices, the so-called *confidence maps*, which suppress the use of image areas showing the object at wrong time steps.

The applications of light fields range from augmented reality to medical imaging. In endoscopic, minimally invasive surgery [12] for instance, a light field of the operation site allows the physician to view the area of interest from any viewpoint without strain to the patient. But since the surroundings during an operation are not static, modeling by dynamic light fields would be appropriate. The method proposed here could, e. g., be used to model the movement of surgical instruments for light field reconstruction during an operation.

Only few articles have been published on the topic of dynamic light fields. The Light Field Video Camera [13] captures moving scenes from different viewing directions using 128 synchronized cameras. The rendering of dynamic light fields from this data was described in [3]. A method for reconstructing a dynamic light field from images of a single camera is introduced in [10]. Here, different movement steps are recorded one after another and registered with each other afterwards. In [1], the time steps for a dynamic light field are defined manually and rendered similarly to [10].

We will demonstrate the applicability of our method on

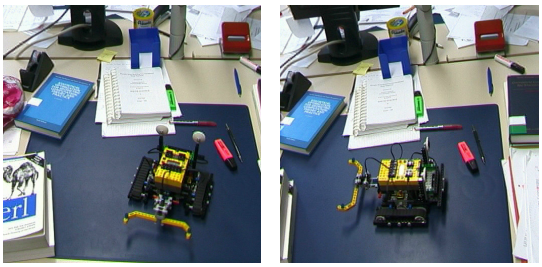


Figure 1: Two images of the *crawler* example sequence.

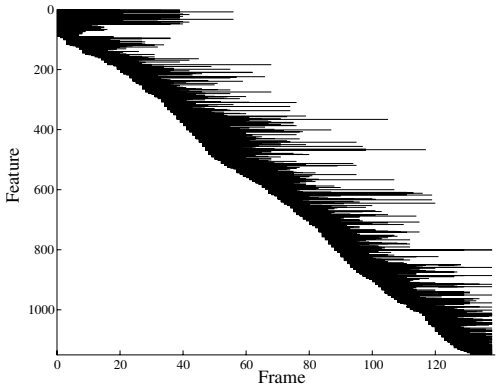


Figure 2: Features tracked on the *crawler* object.

three example image sequences of 139 to 200 frames, each showing a different moving object in front of a static background. The proposed combination of new and known techniques covers for the first time the complete process of reconstructing a dynamic, multibody light field from a single input sequence, as well as its visualization.

2 Multibody Calibration

The starting point of the dynamic light field reconstruction is an image sequence of a scene containing at least one permanently moving but rigid object in front of a static background. Two images of an example for such a sequence are shown in Fig. 1, a toy crawler which moves in a circle on a desk, while the camera is moved back and forth continuously. In the following, the background will be considered as another rigid object, since no distinction can be made a priori between foreground and background. Thus, we assume k objects, where $k \geq 2$.

The following processing steps require the knowledge of point feature correspondences between the images of the sequence. The gradient-based feature detection and tracking method employed here is described in detail in [14].

The average number of frames in which a feature is tracked may be quite low, possibly as few as 10 to 20. As an example, the features found on the moving object in the *crawler* sequence are plotted in Fig. 2. Usually, none of the features is visible throughout the whole sequence. For segmentation, i. e., assigning each feature to one object, and reconstruction using factorization, the features have to be visible in every frame, so that this approach is only practicable for rather short subsequences. Therefore, the succes-

Find initial subsequence with max. number of features	
Segment features into k objects	
Factorize initial subsequence for each object	
	Get next adjacent frame f_i
	Segment new features in f_i, initialized with known features
	Triangulate features using known projections
	Estimate camera parameters for each object
UNTIL every frame calibrated	

Figure 3: Structure chart of the multibody calibration for long image sequences

sive approach illustrated in the structure chart in Fig. 3 is applied and will be explained in the following sections.

2.1 Segmentation and Factorization

For both the segmentation and the factorization process a *measurement matrix* $\boldsymbol{W}$ is created by concatenating the image coordinates of all feature points. As already mentioned, this requires that all feature points are visible in all images. Therefore, the first step is to automatically find the subsequence with the highest number of visible features.

The segmentation is based on the method by Costeira [2] for factorizing scenes with independently moving objects. Two extensions of this algorithm, proposed by Kanatani [6, 7], significantly improve the robustness with respect to noise and are applied here[1]. Segmentation and factorization are performed in two separate, consecutive steps.

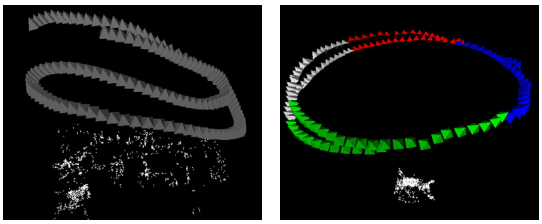
The underlying principle of the segmentation algorithm is that $\boldsymbol{W}$ is of rank 4 in the perspective case for a static 3-D scene, and each additional moving 3-D object increases the rank of $\boldsymbol{W}$ by up to 4. The objects are identified by separating the subspaces of $\boldsymbol{W}$ and thus the features it is composed of. For detailed descriptions of the algorithm we refer to the literature [2, 6, 7].

Once the feature points on each object have been identified, the 3-D structure and camera positions relative to each object are determined. For this purpose, a paraperspective factorization method [9] is applied to each set of features, followed by an iterative non-linear optimization step optimizing in turn the camera pose and 3-D point positions [5].

2.2 Long Image Sequences

As shown in the structure chart in Fig. 3, three main steps after determining the next frame are performed iteratively to calibrate the remaining unknown cameras. First, all features are selected which are visible in at least F_p calibrated frames and which were not considered for the factorization or the preceding iteration, where F_p is smaller than the number of frames for the initial factorization. These are also segmented using the above segmentation algorithm. In order to increase the underlying amount of data, and thus robustness, the already assigned features are used as well,

[1]The source code was kindly provided by the authors at http://www.suri.it.okayama-u.ac.jp/e-program.html



(a) (b)

Figure 4: (a) Reconstruction results of the background of the *crawler* scene. (b) Reconstruction and quantization of the object into four time steps.

and the algorithm is initialized with the known segmentation to decrease the computational cost.

Secondly, the newly found and segmented points are triangulated using the camera parameters of the respective objects. Features with a back-projection error above a threshold are discarded as either erroneous or misclassified.

In the last step the camera pose of the new frame f_i is estimated using the now known 3-D feature points. Every feature in f_i which already has a 3-D correspondence is used to estimate the correct camera parameters by minimizing their back-projection error. The parameters are initialized with the parameters of the preceding camera. As before, the estimation has to be done for each object separately.

These three steps are repeated for all remaining uncalibrated frames. The next frame is chosen alternatingly to be the one before or after the already calibrated subsequence. The results of such a calibration can be seen in Figures 4(a) and 4(b) for the background and the independently moving toy crawler of the sequence of Fig. 1. The reconstructed feature points on the respective objects are visualized as white dots, while the camera poses relative to the background or the crawler, respectively, are depicted as pyramids with their base towards the viewing direction.

3 Light Field Reconstruction and Rendering

In an earlier contribution on dynamic light fields [10], the visualization was based on the assumption that different time steps in scene motion are available. In our case, time steps are equivalent to similar states of object motion. Thus, the goal of reconstruction is to identify and combine images with similar object positions and orientations to individual time steps.

3.1 Time Step Identification

After calibration, the camera motion relative to each object is available and can be used to infer the motion of the object. This camera motion not only depends on the motion of the object, but it also includes the motion of the camera itself. In order to get the real motion relative to the object, the camera's own motion has to be eliminated.

Since no common world coordinate system is available, the reconstruction for each object will differ from the others by an arbitrary rotation, translation and scaling. This issue has not been addressed in [2], but it was encountered likewise for the dynamic light fields in [10].

The object containing the most features is selected as the background of the scene. Assuming that the poses of the first background camera $\boldsymbol{P}_{0,1}$ and the first camera of any object $\boldsymbol{P}_{i,1}$ are the same, any object camera can be transformed to the background coordinate system:

$$\boldsymbol{P}'_{i,j} = \boldsymbol{P}_{i,j}\boldsymbol{M}_{i,1}^{-1}\boldsymbol{M}_{0,1}\,, \tag{1}$$

where $\boldsymbol{M}_{i,j}$ is a 4×4 extrinsic camera parameter matrix for object i and camera j. It is built from the rotation $\boldsymbol{R}_{i,j}$ and translation $\boldsymbol{t}_{i,j}$ of the respective camera. A camera parameter matrix $\boldsymbol{P}_{i,j}$ is thus composed of

$$\begin{aligned}\boldsymbol{P}_{i,j} &= (\boldsymbol{K}_j|\boldsymbol{0}_3)\boldsymbol{M}_{i,j} = \\ &= (\boldsymbol{K}_j|\boldsymbol{0}_3)\begin{pmatrix} \boldsymbol{R}_{i,j}^T & -\boldsymbol{R}_{i,j}^T\boldsymbol{t}_{i,j} \\ \boldsymbol{0}_3^T & 1 \end{pmatrix}.\end{aligned} \tag{2}$$

$\boldsymbol{K}_j$ is the 3×3 intrinsic camera parameter matrix for camera j.

The inverse transformation is applied to each (homogeneous) 3-D object point. The remaining scale factor is determined by assuming that the 3-D points should be at the same distance from the cameras. Therefore, the scaling is calculated as the ratio between the distances of the centers of mass of the 3-D point clouds of object and background, and again applied to each camera and point.

Both background and object reconstruction are now in the same coordinate system, although the transformation may not be exact since the scaling is calculated only by a heuristic measure. An accurate calculation of the scale factor will be subject to future work. The object-relative camera movement is now calculated as the transformation between the positions of two corresponding cameras $\boldsymbol{M}_{0,j}$ and $\boldsymbol{M}'_{i,j}$, transformed back to the common coordinate system by $\boldsymbol{M}_{0,1}$:

$$\boldsymbol{P}''_{i,j} = (\boldsymbol{K}_j|\boldsymbol{0}_3)\boldsymbol{M}_{0,1}\boldsymbol{M}_{0,j}^{-1}\boldsymbol{M}'_{i,j}\,. \tag{3}$$

From these corrected camera matrices the similar object positions are calculated by applying a vector quantizer to the camera position vectors. The desired number of time steps can be specified and the camera positions are grouped around a codebook vector for each step, minimizing the intra-class distance. An example for the resulting quantization is shown in Fig. 4(b) for the *crawler* sequence. Here, the camera positions are subdivided into four time steps.

3.2 Rendering

By separating the resulting time steps into one static light field each, the rendering can be done again like in [10] by enabling the renderer to switch back and forth between the light fields. However, this approach has the drawback that only a fraction of the images can be used for each time step.

The new rendering technique we propose is based on the *Unstructured Lumigraph* [1], but it is applicable to other renderers as well. So-called *confidence maps* are calculated for each image in the sequence and for each time step. They contain information about which parts of an image are to be used for which time step. The confidence map may contain three different values, e. g., 0 for the foreground object if it is invisible, 2 if it is visible, and 1 for all background pixels.

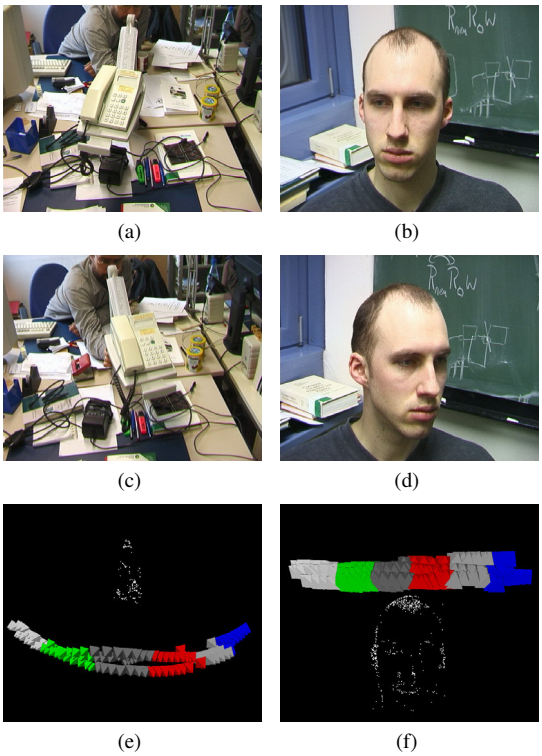


Figure 5: Two images each of the *phone* (a, c) and *head* (b, d) example sequences. (e) *Phone* and (f) *head* object reconstruction and quantization of six time steps each.

Whether a pixel in an image belongs to the background or the object is determined by the following procedure: The segmented feature points in each image are connected by a mesh using delaunay triangulation [11]. A triangle is assigned to a moving object if at least one of its vertices belongs to the object. Otherwise it is allocated to the background.

During rendering, contributions from different images are weighted according to these confidence values. Thus, if several images contribute to a rendered patch, pixels from the object at a wrong time step are discarded, while pixels from the object at the correct time step will always overlay background pixels from other images. This ensures that for rendering the background, all images can be used for every time step, and a selection of images is only necessary for the moving object. Rendering different time steps is done by switching from one set of confidence maps to another.

4 Experiments

As examples for the reconstruction of long image sequences, three real sequences showing different moving objects in front of a static background were chosen. Beside the *crawler* sequence of Fig. 1, the examples show a rotating telephone arm (Figures 5(a) and 5(c)) and a person turning his head from left to right (Figures 5(b) and 5(d)). A prerequisite for the reconstruction to work is that enough features are found on each object. The examples were selected accordingly.

The total number of frames in the sequences ranged from 139 to 200, but the initial factorization was done on 10 (*crawler*) to 35 (*head*) frames only, depending on the size of the moving object. The total number of features assigned to background and object can be seen in Table 1, as well as the average number of point correspondences used on the object. Each feature had to be visible in at least 8 frames to be used for 3-D reconstruction.

sequence	frames	feat. bg	feat. obj	corr. obj
crawler	139	2117	1150	16.7
phone	145	2198	234	51.8
head	200	1172	718	88.1

Table 1: Some statistics on the example sequences: total number of frames (2nd column), total number of features on background (3rd column) and object (4th column), and average number of point correspondences per feature found on the object (5th column).

The final result of the object reconstructions for each sequence are depicted in Figures 4(b), 5(e) and 5(f), respectively. Here, the final camera path is visible which results from deducting the camera's own motion. The quantization by camera position, as described in Sect. 3.1, is illustrated by different shades for each "time" step.

Figure 6 shows four images rendered from the resulting light fields for each of the three sequences, using the rendering method introduced in Sect. 3.2. For all four images of each sequence, the camera pose was the same, which is reflected by the identical background in each image, but the object was rendered for four different time steps, and is thus at different positions. Note that the camera poses for the rendered images were not part of the original sequence, but chosen arbitrarily. For the *crawler* light field the image sequence was subdivided into eight time steps, while the other two light fields consist of six time steps each.

5 Conclusion

In this contribution we proposed a solution for reconstructing a dynamic light field of a scene including at least one rigidly moving object. Prior to factorizing an initial subsequence for the background and each moving object separately, a motion segmentation algorithm is applied to automatically acquired features. The calibration is extended to the whole sequence by alternatingly triangulating and segmenting additional features and calibrating new frames. For the final light field the resulting 3-D reconstructions for each object are merged into a common coordinate system and a common scaling is approximated. The camera positions are then divided into different "time steps" of similar camera positions. Rendering is done by masking the moving objects in the original images that belong to time steps which are currently not observed using *confidence maps*, while the static background is used from every image.

Although this method already constitutes an improvement over an earlier system for dynamic light field reconstruction [10], many further developments are possible. The rendering quality is still limited by the precision of the object segmentation in the original images. However, many improved segmentation algorithms exist which are better

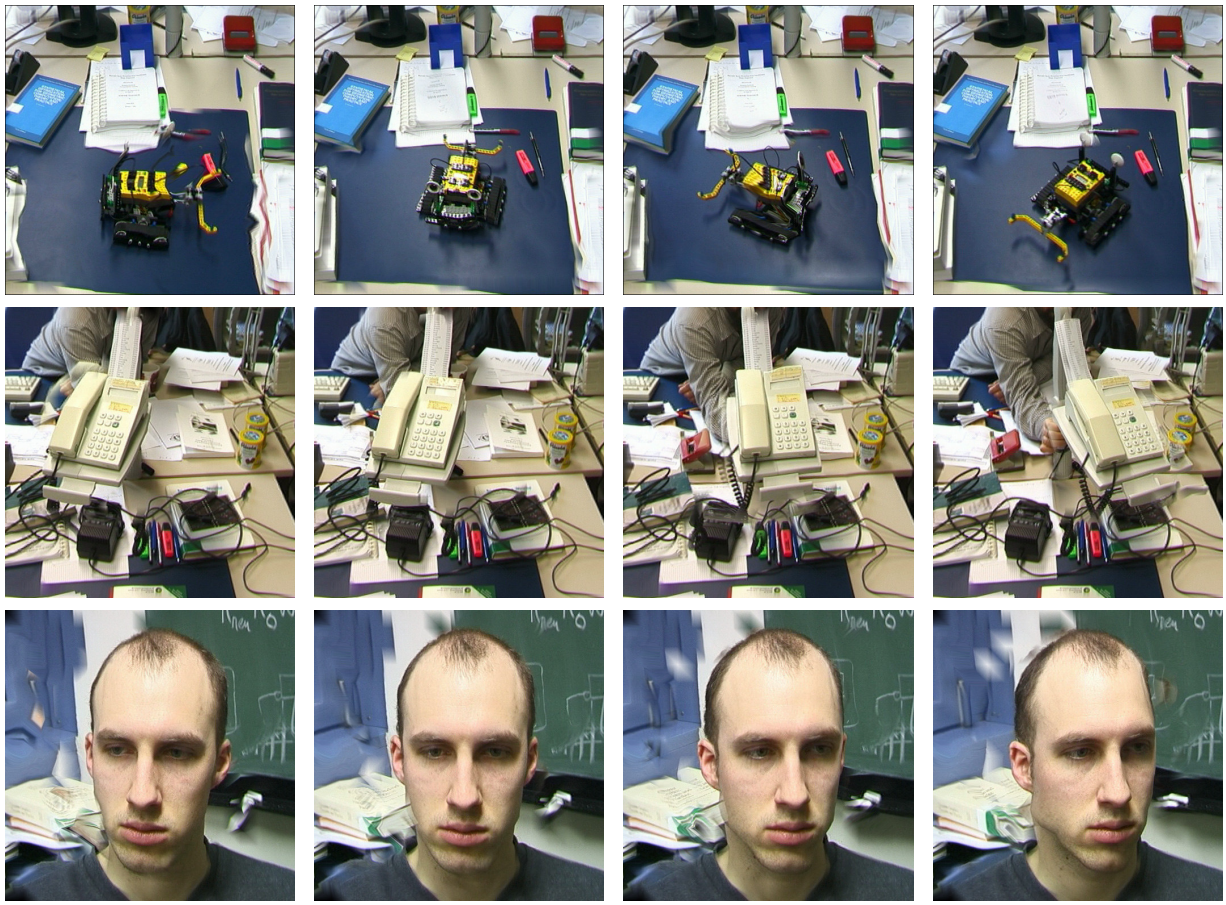


Figure 6: Rendered images for four different time steps of the *crawler* (top row), *phone* (middle row) and *head* sequence (bottom row), seen from the same camera position.

suitable for this task than the currently used one. In reconstruction, the next step will be to consider cases of disrupted motion, disappearing and new objects. Calculating the true scale factor between the different reconstructions of each object remains an open problem.

References

[1] C. Buehler, M. Bosse, L. McMillan, S. J. Gortler, and M. F. Cohen. Unstructured lumigraph rendering. In *Proc. of ACM SIGGRAPH 2001*, pages 425–432. ACM Press, August 2001.

[2] J. P. Costeira and T. Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[3] B. Goldlücke, M. Magnor, and B. Wilburn. Hardware-accelerated dynamic light field rendering. In *Vision, Modeling and Visualization*, pages 455–461. infix, November 2002.

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

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

[6] K. Kanatani. Motion segmentation by subspace separation and model selection. In *8th International Conference on Computer Vision*, volume 2, pages 586–591, Vancouver, Canada, July 2001.

[7] K. Kanatani and Y. Sugaya. Multi-stage optimization for multi-body motion segmentation. In *Proc. Australia-Japan Advanced Workshop on Computer Vision*, pages 25–31, Adelaide, Australia, September 2003.

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

[9] C. J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):206–218, March 1997.

[10] I. Scholz, J. Denzler, and H. Niemann. Calibration of real scenes for the reconstruction of dynamic light fields. *IEICE Transactions on Information & Systems*, E87-D(1):42–49, January 2004.

[11] J. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications*, 22(1-3):86–95, 2002.

[12] F. Vogt, S. Krüger, J. Schmidt, D. Paulus, H. Niemann, W. Hohenberger, and C. H. Schick. Light Fields for Minimal Invasive Surgery Using an Endoscope Positioning Robot. *Methods of Information in Medicine*, 43:403–408, 2004.

[13] B. Wilburn, M. Smulski, H.-H. Kelin Lee, and M. Horowitz. The light field video camera. In *Proc. Media Processors, SPIE Electronic Imaging*, pages 29–36, 2002.

[14] T. Zinßer, C. Gräßl, and H. Niemann. Efficient feature tracking for long video sequences. In *Pattern Recognition, 26th DAGM Symposium*, pages 326–333, Tübingen, Germany, August 2004. Springer-Verlag, Berlin.