Heinrich Niemann and Ingo Scholz

Evaluating the Quality of Light Fields Computed from Hand-held Camera Images

appeared in:
Pattern Recognition Letters
Volume 26(3), February 2005, In Memoriam: Azriel Rosenfeld
pp. 239–249

Evaluating the Quality of Light Fields Computed from Hand-held Camera Images

Heinrich Niemann [a], Ingo Scholz [a,1]

[a] *Lehrstuhl für Mustererkennung, Martensstr. 3, 91058 Erlangen, Germany*

Abstract

Given an image sequence recorded by a hand-held camera we examine the computation of a light field without any further input data. Using structure-from-motion algorithms and optimization techniques camera motion and a 3-D reconstruction of the scene are established. The light field is completed by computing local depth information for each input image. During experimental evaluation a special focus is set on the effects of falsely estimated intrinsic parameters as well as different depth representations on the quality of the resulting light fields.

Key words: Camera calibration, structure-from-motion, image-based rendering

In Memoriam

At the time of ICPR 1982 in Munich, Prof. Azriel Rosenfeld was president of IAPR. He gave suggestions and advice for the scientific program, and he presented an invited talk entitled "Image Analysis: Progress, Problems, and Prospects". He said "More important, during the past few years the field has begun to develop a scientific basis" – to which he contributed essential aspects. Among those contributions was the transition from the 2D image to the 3D scene. This paper deals with recovering 3D structure from a time sequence of 2D images recorded by an uncalibrated camera.

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

The picture shows A. Rosenfeld with the general chairman of ICPR 1982, H. Marko, the Bavarian Minister of Economy, A. Jaumann, and the president of the Technical University of Munich, W. Wild (from right to left)

1 Introduction

The field of computer graphics has undergone rapid development in the past years, leading to the ability of rendering images which are often indiscriminable from real photographs. Nevertheless, achieving such results is a time-consuming task which requires even artistic skills. On the other hand, real-time rendering of complex environments requires high computational power and photo-realistic quality is still unachievable.

Such traditional approaches to rendering rely on scene models consisting of geometric primitives covered by textures. Recovering geometry from real scenes is difficult, at best, which, in recent years, has given rise to the concept of image-based modeling and rendering with the most prevalent representations being the light field of Levoy and Hanrahan (1996) and the lumigraph, proposed by Gortler et al. (1996). Although the two approaches have been developed independently from each other, they apply many similar concepts and nowadays the term "light field" is more often used for combinations of them, as it will be throughout this contribution. Their advantage over geometry-based modeling lies in their use of images of real scenes as input data which simplifies photo-realistic rendering of these scenes.

Nevertheless, the correct and robust computation of a light field from a set

of images, while keeping the costs low, is not an easy task. Using a camera gantry is a cost-intensive solution and often very inflexible. An inexpensive, yet flexible, alternative is the use of a single hand-held camera, and the application of structure-from-motion algorithms to the recorded image sequence. For some applications such as laparoscopic surgery, cf. Vogt et al. (2004), a single hand-held camera is even the only feasible method. Computing the information required for a light field from such an uncalibrated image sequence poses some serious problems. It is desirable that the process be as stable as possible, but given noisy input data self-calibration algorithms for computing intrinsic camera parameters often lack this requirement. In this contribution we will address the processing steps for computing a light field, including a robust method for estimating camera parameters which is mainly based on non-linear optimization techniques. Additionally, we will evaluate the impact of an inaccurate intrinsic parameter estimation on light field quality.

1.1 Light Fields

The idea of the light field was derived from the *plenoptic function* introduced by Adelson and Bergen (1991). It describes the appearance of a volume in space using seven parameters, i. e. the viewpoint of the observer in world coordinates, the two angles of the viewing direction and the wavelength of the observed light ray at a certain time. The light field breaks down this high dimensional space using several restrictions. The observed scene is assumed to be constant over time, and instead of the intensity for every wavelength only one color value is modeled, thus removing two parameters. In addition to that, the air between the observer and the scene surface is assumed to be transparent so that the intensity of the light ray emitted from a surface point in one direction stays constant, no matter where the observer is located on this light ray. By selecting a suitable parameterization the plenoptic function is thus reduced to four parameters.

Both Levoy and Gortler use a two-plane parameterization to represent the light field, representing each light ray by one point on each plane. Here, the cameras are placed on a regular grid on one plane, while the other plane is the common focal plane of the cameras. In the following many more parameterizations have been proposed, a summary of which can be found in Schirmacher et al. (2001). Nevertheless, the goal has always been to reduce the restrictions of the two-plane model. The first *free form* light field renderer was introduced by Heigl et al. (1999) which allowed the placement of the cameras at almost arbitrary positions in space. The most generalized light field model so far is the *Unstructured Lumigraph* introduced by Buehler et al. (2001).

Since light fields may consist of hundreds of input images, an important issue

Detect and track point features in every image $\boldsymbol{f}_i, i = 1, \ldots, N$ (Sect. 2.1)
Select sub-sequence with most features visible in all images $\boldsymbol{f}_s, s = i_a, \ldots, i_b$ (Sect. 2.2)
Apply paraperspective factorization method to sub-sequence (Sect. 2.2)
Optimize camera parameters and 3-D points non-linearly (Sect. 2.2)
Compute camera parameters of remaining images $\boldsymbol{f}_r, r = 1, \ldots, i_a - 1, i_b + 1, \ldots, N$ by non-linear optimization (Sect. 2.3)
Generate depth map or local proxy for each image $\boldsymbol{f}_i$ (Sect. 3)

Fig. 1. Processing steps for computing a light field from an uncalibrated image sequence of N images

in image-based rendering is the storing and compression of these large amounts of data. Knowledge of 3-D geometry, as it is often the case for light fields, has been exploited before for coding of video sequences and was adapted likewise for light field compression as in Magnor et al. (2003). However, compression will not be treated here as it would exceed the bounds of this contribution.

1.2 Outline

The focus of this article is on the computation of light fields as opposed to their rendering, therefore the latter will be discussed only marginally. The general processing sequence is given in Figure 1, where for each step the corresponding section is indicated. Thus, Section 2 covers the process of computing the camera motion parameters from only the uncalibrated input image sequence. The employed methods of feature tracking, structure-from-motion using paraperspective factorization, and non-linear optimization are explained in detail. The resulting light field is only sparsely sampled which can lead to rendering artifacts in areas containing significant depth discontinuities. Section 3 therefore addresses the computation of depth maps and 3-D meshes, called geometric proxies, for modeling the scene structure. Experimental results obtained with these procedures on both synthetic and real data are presented in Section 4, and a summary will be given in the conclusion, in Section 5.

2 Camera Parameter Computation

In essence, a light field is only a sampling of the plenoptic function where each image contributes a set of light rays. In order to know which light rays are added by a certain image the position, orientation and internal parameters of the recording camera have to be available. The process of computing these parameters, which will be introduced in the following, is based on the work of Heigl (2004), although in this investigation, it is reduced to robust parameter

estimation using non-linear optimization, and new modules such as a reliable feature tracking were included.

2.1 Feature Selection and Tracking

Most algorithms for camera motion estimation require a set of point correspondences in two or more images. For light fields computed from image sequences of a hand-held camera features often have to be tracked through hundreds of images, at times with low quality or varying illumination. Therefore, a highly robust and accurate feature tracking system is required. In most cases an extension of the tracker by Tomasi and Kanade (1991) is applied, in our case the system of Zinßer et al. (2004). It combines many extensions of the original tracking algorithm such as affine motion estimation, resolution hierarchies, linear illumination compensation and feature drift prevention.

2.2 Factorization of an Initial Sequence

Given an uncalibrated camera many methods have been proposed to obtain its motion and internal parameters. The basis of the procedure described in the following two sections is the method introduced by Hartley (1994) for recovering a Euclidean reconstruction by Levenberg-Marquardt (LM) optimization. For a camera pose in world coordinates we denote the position of its optical center as $\boldsymbol{t}_i$ and its rotation as $\boldsymbol{R}_i$, where the columns of $\boldsymbol{R}_i$ contain the coordinate axes of the camera coordinate system. The projection of a 3-D point $\boldsymbol{p}_j$ onto an image point $\boldsymbol{q}_{i,j}$ in image $\boldsymbol{f}_i$ is given as

$$\boldsymbol{q}_{i,j} = \boldsymbol{P}_i \boldsymbol{p}_j = \boldsymbol{K}_i (\boldsymbol{R}_i^T | - \boldsymbol{R}_i^T \boldsymbol{t}_i) \boldsymbol{p}_j \quad , \tag{1}$$

$(\cdot|\cdot)$ being a concatenation of two matrices, where

$$\boldsymbol{K}_i = \begin{pmatrix} f_x & \beta & u \\ 0 & f_y & v \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

contains the intrinsic parameters of the ith camera viewpoint, i. e. the principal point (u, v), the focal length f_x and f_y along the coordinate axes of the sensor, and the coordinate axes skew β. Both $\boldsymbol{q}_{i,j}$ and $\boldsymbol{p}_j$ are given in homogeneous coordinates. The parameters of $\boldsymbol{K}_i$, $\boldsymbol{R}_i$ and $\boldsymbol{t}_i$ are estimated by minimizing the back-projection error of each 3-D point. For an estimated projection $\widehat{\boldsymbol{q}}_{i,j} = \widehat{\boldsymbol{P}}_i \widehat{\boldsymbol{p}}_j$ the error is calculated as the difference to the corresponding true image

feature $\boldsymbol{q}_{i,j}$,

$$\boldsymbol{\epsilon}_{i,j} = \boldsymbol{q}_{i,j} - \widehat{\boldsymbol{q}}_{i,j} \quad . \tag{3}$$

The total back-projection error for image $\boldsymbol{f}_i$ is thus defined as $\boldsymbol{\epsilon}_i^T \boldsymbol{\epsilon}_i$, $\boldsymbol{\epsilon}_i$ being the concatenation of all $\boldsymbol{\epsilon}_{i,j}$ in a single column vector. In contrast to Hartley, who optimizes the 3-D points $\widehat{\boldsymbol{p}}_j$ together with the camera parameters, points and camera parameters are estimated in turn to reduce the parameter space considerably.

As stated by Hartley (1994) this optimization requires a good initialization in order to work correctly. Having multiple images and a large number of point correspondences it is obvious to choose a factorization method for this task. Tomasi and Kanade (1992) introduced this method for estimating structure and motion of a sequence assuming orthographic or weak-perspective projection. By combining all point correspondences in one measurement matrix and decomposing it using singular value decomposition (SVD) the camera poses and 3-D point positions are derived simultaneously. A better approximation of reality is the paraperspective projection model which is assumed by the extension of Poelman and Kanade (1997).

This factorization is applied to a suitable sub-sequence of the input image sequence since it requires that every feature be visible in every image. Usually the longest sub-sequence with a certain minimum number of features is chosen. The result is supplied as initialization to the Levenberg-Marquardt optimization of the back-projection error $\boldsymbol{\epsilon}_i^T \boldsymbol{\epsilon}_i$ introduced before.

The main problem at this point is that the parapespective factorization is not able to supply estimates for the intrinsic camera parameters. A commonly used solution is to apply self-calibration like the estimation of the absolute quadric as proposed by Triggs (1997). Its drawback is that for noisy point correspondences these procedures often lack robustness. Therefore, the intrinsic parameters are chosen arbitrarily, but as close to the real values as possible, and kept constant over the whole sub-sequence. Thus the skew factor β is set to zero, the principal point (u, v) is set to the image center and the focal length is set to a sensible value for the camera used with $f_x = f_y$. For common video cameras the choice of the first two parameters is usually quite close to the truth. Considering the focal length the important question is how a wrong estimate affects the quality of the light field. This issue will be assessed in Section 4.

2.3 Extension to Long Image Sequences

The result of the above computation is a scene and motion reconstruction for a short sub-sequence of the complete input image sequence. In the following

step this reconstruction is extended to the rest of the image sequence using the same optimization technique already applied above. For each image adjacent to the known sub-sequence new 3-D points are triangulated and non-linearly optimized if they are visible in a sufficient number of known images. As initialization for the projection matrix the parameters of the closest known camera are used. The LM optimization proved to be robust enough to converge to the correct parameters in most cases. Even an initialization using a linear prediction of the camera position and rotation – which is sensible since we assume a smoothly moving hand-held camera – showed no significant improvement.

The problem of estimating the correct intrinsic parameters persists for this extension step. Like in Section 2.2 one possible solution is to keep the intrinsic parameters constant regardless of their true values. The optimization will compensate this by changing the distance of the camera positions to the scene, a procedure which nevertheless proved to process the whole sequence very reliably.

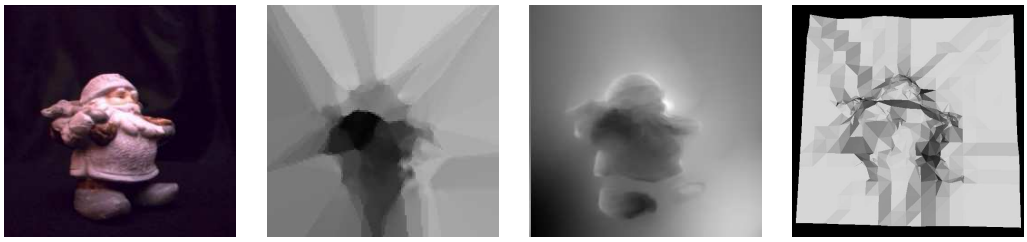
The alternative is to additionally estimate some or all of the intrinsic parameters. In this case there are 7 to 10 variable parameters per camera pose instead of only 6, so that the process is necessarily more unstable. Depending on feature quality, the focal length and the distance from the scene may still be confused and thus wrongly estimated. The success of the intrinsic parameter estimation depends on the scene configuration and camera movement. An experimental investigation of this issue follows in Section 4.

3 Light Field Reconstruction

In case of a dense sampling of the scene as it is done by Levoy and Hanrahan (1996) the light field computation is complete as soon as the camera parameters are known. The required sampling density for preventing aliasing or ghosting artifacts was analyzed by Chai et al. (2000). For a sparse sampling when using a hand-held camera a correct light field representation requires additional depth information. It can be supplied either as dense depth maps or as a geometric proxy, the computation of these alternatives will be introduced in the following.

3.1 Depth Maps

Computing a dense depth map for each of the input images using the 3-D points obtained during scene and motion reconstruction is straight-forward and done as follows. The 2-D features in an image with known 3-D position



(a) (b) (c) (d)

Fig. 2. Different depth representations of image (a): (b) interpolated from 3-D points, (c) variational approach and (d) a local proxy (see Section 3.2)

yield a sparse depth map which can be filled by interpolating the depth between the nearest neighbours of each pixel. The result of such a simple and fast depth map computation is shown in Figure 2(b) for the input image of Figure 2(a).

The computation of a depth or disparity map from stereo images has been investigated in numerous publications and may lead to better results than the method above, although often at a higher computational cost. As an example we will use depth maps generated by a variational approach by Alvarez et al. (2002) as seen in Figure 2(c). It makes use of the fundamental matrix $\boldsymbol{F}$ for disparity estimation between two images with camera projection matrices $\boldsymbol{P}_i$ and $\boldsymbol{P}_j$ which is readily available from the preceding camera parameter reconstruction. If $\boldsymbol{P}_i$ (and similarly $\boldsymbol{P}_j$) is decomposed into $\boldsymbol{P}_i = (\boldsymbol{X}_i|\boldsymbol{x}_i)$ with $\boldsymbol{X}_i \in \mathbb{R}^{3\times3}$ and $\boldsymbol{x}_i \in \mathbb{R}^3$, $\boldsymbol{F}$ is computed as

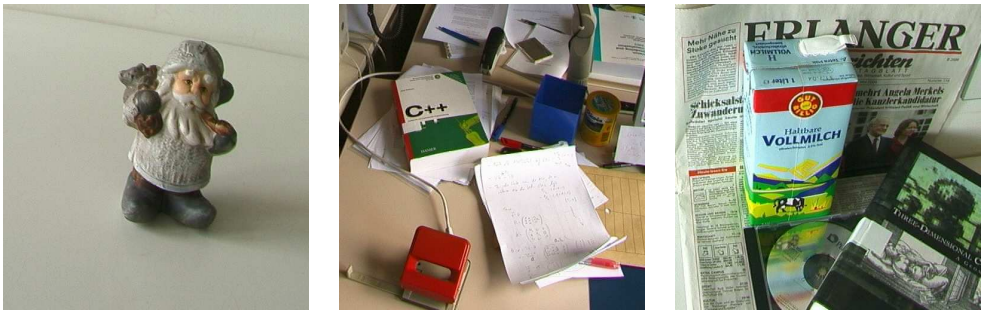
$$\boldsymbol{F} = [\boldsymbol{x}_j - \boldsymbol{X}_j\boldsymbol{X}_i^{-1}\boldsymbol{x}_i]_\times(\boldsymbol{X}_j\boldsymbol{X}_i^{-1}) \quad . \tag{4}$$

$[\boldsymbol{a}]_\times$ denotes the antisymmetric matrix performing an outer left multiplication by $\boldsymbol{a}$:

$$[\boldsymbol{a}]_\times = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \quad . \tag{5}$$

3.2 Geometric Proxies

The term *geometric proxy*, for representing the geometric properties of a scene, was introduced by Buehler et al. (2001). Instead of a depth map for each image depth information is provided as a global geometric model which allows much higher frame rates for rendering on graphics hardware. In our case, geometry information is only available as a point cloud which is not necessarily globally consistent because of error accumulation. For long sequences the same feature may be present multiple times at different positions which complicates the



(a) (b) (c)

Fig. 3. Example images from sequences (a) *santa1*, (b) *desk1* and (c) *milk*

computation of a single geometry model. Therefore, the geometry is supplied as one triangular mesh for each image generated from the visible 3-D points. An example for such a local proxy is shown in Figure 2(d).

4 Experimental Results

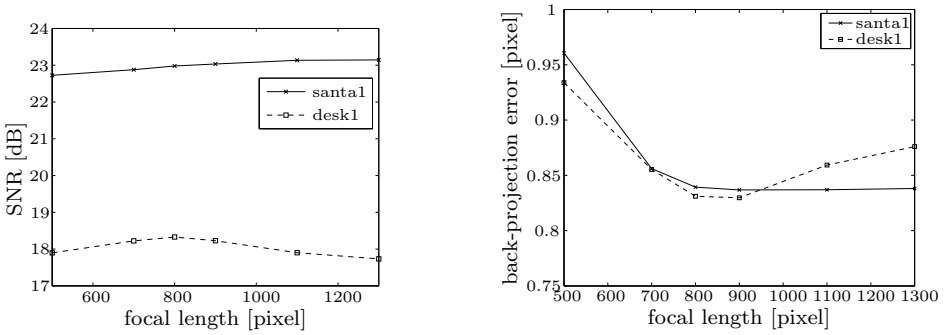
In the following, experiments will be presented regarding the choice or estimation of focal length, as discussed in Section 2, and the type of depth information used during rendering. In order to assess the correctness of a reconstruction the usual measure applied is the back-projection error. However, this error is not necessarily reflected in the quality of the resulting light field, and it is not applicable for the comparison of depth maps.

Therefore, a new method for evaluating light fields is introduced. For each image $\boldsymbol{f}_i$ in the original sequence the corresponding view is rendered from the light field using the calculated camera parameters. The original image in question is not used for rendering so that the rendered image $\boldsymbol{g}_i$ is only composed from neighbouring images. By computing the difference image $\boldsymbol{d}_i = \boldsymbol{f}_i - \boldsymbol{g}_i$ the result can be visualized, but in the following the average signal-to-noise ratio will be used which is computed as

$$SNR = \frac{1}{N}\sum_{i=1}^{N}(10\log_{10}(\bar{\boldsymbol{f}}_i/\bar{\boldsymbol{d}}_i)) \tag{6}$$

where $\bar{\boldsymbol{f}}_i$ and $\bar{\boldsymbol{d}}_i$ are the mean of the squared pixel values of image $\boldsymbol{f}_i$ and $\boldsymbol{d}_i$ respectively.

For the experiments six image sequences were taken with a hand-held camera and light fields computed from them. The sequences are named *santa1* (207 images), *santa2* (155 images), *desk1* (105 images), *desk2* (147 images), *desk3* (179 images) and *milk* (190 images) and consist of color images sized 512×512 pixels. For sequence *santa2* the camera was moved once in a complete circle



(a) (b)

Fig. 4. Signal-to-noise ratio and back-projection error depending on the choice of focal length for sequences *desk1* and *santa1*

around the scene (cf. Figure 7), whereas for the other sequences the camera was moved in a zig-zag motion in front of the scene with a viewing angle of about 45° as depicted e. g. in Figure 5. Example images of three of the sequences are shown in Figure 3, the other three sequences are similar. The light fields were generated according to the processing steps in Figure 1 using the interpolated depth maps as seen in Figure 2(b). The renderer used for SNR computation is an implementation of the *Unstructured Lumigraph* by Buehler et al. (2001).

4.1 Focal Length Selection

The computation of the intrinsic parameters of a camera from image features, called self-calibration, is a very difficult task for noisy input data. In Section 2.2 the question was raised whether wrongly estimated parameters have a significant influence on the quality of a light field. This issue is investigated by reconstructing two image sequences, *desk1* and *santa1*, several times with different preset focal length parameters. The results for the back-projection error and signal-to-noise ratio are plotted in Figure 4. The true focal length of the camera was calibrated as $f_x = 844$ and $f_y = 924$, but for the experiments f_x and f_y were assumed equal and ranged from 500 to 1300 pixels.

The experiments showed that indeed back-projection error and SNR are best for the true focal length. Nevertheless, with about 0.15 pixel and 0.6 dB variation respectively the error using a wrong focal length is quite low, which leads to the conclusion that knowledge of the true focal length is desirable although light field quality suffers only little if it is not available.

The image sequences *desk2*, *desk3* and *santa2* were generated to test a varying focal length during the extension step of Section 2.3. In the first two sequences the focal length was changed during recording while the camera was kept at the same distance to the scene. Moving the camera around an object proved to

Table 1
SNR and back-projection error (bpe) for sequences with changing focal length

Sequence	*desk2*			*desk3*			*santa2*	
intr. params.	none	all	f_x, f_y	none	all	f_x, f_y	none	f_x, f_y
bpe (pixel)	1.10	0.88	0.93	1.22	0.98	0.99	1.14	1.15
SNR (dB)	15.9	17.0	16.0	15.7	15.3	15.6	13.0	13.7

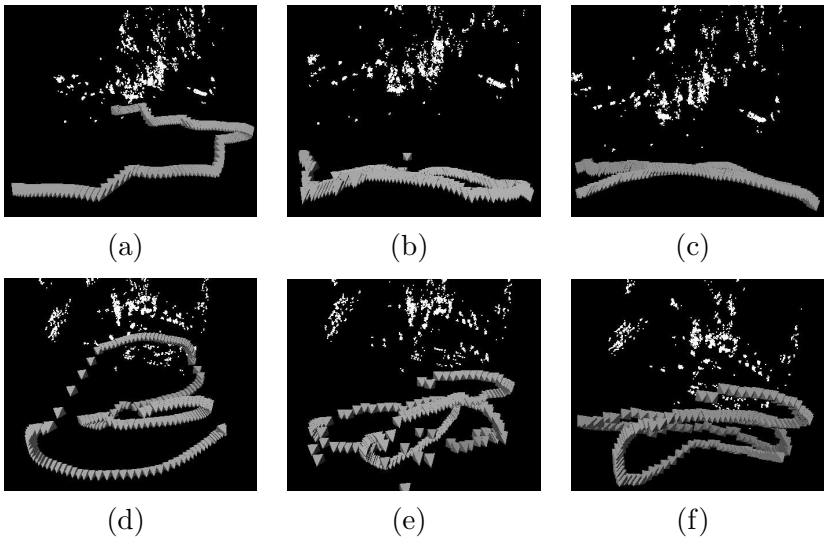


Fig. 5. Camera parameter and 3-D point reconstruction results for varying focal length during recording of sequences *desk2* (a – c) and *desk3* (d – f). Left column: fixed intrinsic parameters. Middle: all parameters estimated. Right: f_x and f_y estimated. Each camera is depicted as a pyramid with its tip the camera center and its base the image plane.

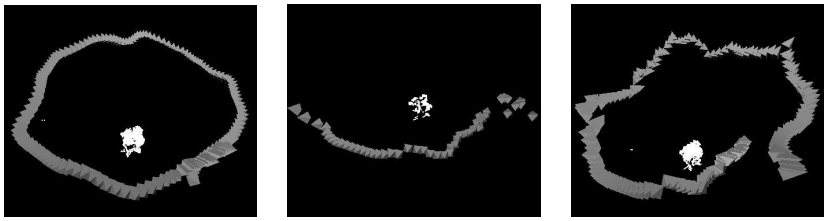
be a more difficult configuration which was examined using the third sequence, *santa2*.

For each image sequence three different light fields were computed, the first with assuming a constant focal length for the whole sequence (column "none" in Table 1), the second with estimation of all intrinsic parameters (f_x, f_y, u and v) during the extension step of Section 2.3 (column "all"), and the third with a fixed principal point (u, v) but estimating f_x and f_y (column "f_x, f_y").

The results for the first two sequences are depicted in Figure 5, images (a) to (c) showing the camera pose and 3-D point reconstructions of each of the three light fields for sequence *desk2*, and (d) to (e) for sequence *desk3*. It can be seen that for constant focal length the algorithm compensated the zoom variation by changing the distance of the camera to the scene instead. When estimating all intrinsic parameters for each image the reconstruction was successful as well, although the number of outliers visible in Figures 5(b) and 5(e) indicates the reduced stability of the estimation. Estimating only f_x and f_y is apparently

(a) (b)

Fig. 6. Rendered images of light fields from sequence *desk3*, (a) computed with constant intrinsic parameters (cf. Table 1, column "none"), (b) with all intrinsic parameters (cf. Table 1, column "all").



(a) (b) (c)

Fig. 7. Reconstructions of the *santa2* sequence with (a) constant and (b) completely variable intrinsic parameters, and (c) only variable focal length

a good compromise between accuracy and robustness, as for both sequences, shown in Figures 5(c) and 5(f), the drawbacks of estimating fewer or more parameters are not evident.

The resulting back-projection error and SNR are given in Table 1. Again it can be said that although estimating the intrinsic parameters results in a lower back-projection error, it does not necessarily increase the quality of the light field according to the SNR. This impression is supported by the two rendered images seen in Figure 6 from the *desk3* light fields without (left) and with (right) estimating the intrinsic parameters. Nevertheless, estimating only the focal length f_x, f_y seems to be a viable compromise.

As Heigl (2004) already stated, estimating the intrinsic parameters in a sequence while the camera moves around the scene usually causes the extension step to fail as it can be seen in Figure 7(b) for sequence *santa2*. As before, estimating only f_x and f_y proved to be a more reliable compromise as it yielded a complete reconstruction as shown in Figure 7(c). Nevertheless, back-projection error and SNR of Table 1 do not suggest one method to be superior to the other, especially since such a sparsely sampled light field is generally of a low quality.

Table 2
Comparison of different depth information types

Sequence	depth type	SNR	time
santa1	3-D points	23.12 dB	0.87 s
santa1	variational	23.19 dB	0.79 s
santa1	local proxies	20.70 dB	0.26 s
milk	3-D points	12.18 dB	0.83 s
milk	variational	12.28 dB	0.82 s
milk	local proxies	10.88 dB	0.25 s

(a) (b) (c)

Fig. 8. One close-up view from the *santa1* light field using (a) interpolated depth maps, (b) variational approach and (c) local proxies

4.2 Depth Map Comparison

For sparsely sampled light fields as they are computed from images of a hand-held camera a correct depth map is very important for the correct rendering of images. In the previous section the light fields were created using depth maps interpolated from reconstructed 3-D points. In the following the three types of depth information introduced in Section 3 will be compared.

For two image sequences, *santa1* and *milk*, all three depth maps (cf. Figure 2) were computed and the results compared in Table 2. For the signal-to-noise ratio, the evaluation shows no obvious advantage of one of the two depth map types over the other, although they both perform better than the local proxies. Nevertheless, the rendering time for one image is divided by three when using local proxy information, underlining at least their superiority in computational efficiency. For comparison three close-up views rendered from the *santa1* light field as seen from the same camera pose but using different depth information are given in Figure 8.

5 Conclusion

In this contribution we have given an overview over the process of computing a light field from an image sequence recorded by a hand-held camera. By applying structure-from-motion techniques like a factorization method and non-linear optimization the camera pose is reconstructed for each image in the sequence. Combining the motion parameters with 3-D or depth information yields a sparsely sampled light field of the recorded scene.

Estimating intrinsic camera parameters in addition to camera pose often reduces the robustness of the reconstruction process. Therefore, the influence of inaccuracies in the estimation of intrinsic camera parameters on the quality of the light field was examined in several experiments, and different types of depth information were assessed in the same regard. For these evaluations a new method for measuring the quality of a light field was introduced which yields an average signal-to-noise ratio. In the investigated case it corresponds better to subjective image quality than the back-projection error. However, this issue requires further analysis.

Acknowledgements

Thanks to Luis Alvarez and Javier Sánchez for providing the source code for the variational disparity map algorithm and to Christian Vogelgsang for the unstructured lumigraph renderer using local proxies.

References

Adelson, E. H., Bergen, J. R., 1991. Computational Models of Visual Processing. MIT Press, Cambridge, MA, Ch. 1 (The Plenoptic Function and the Elements of Early Vision).

Alvarez, L., Deriche, R., Sánchez, J., Weickert, J., March/June 2002. Dense disparity map estimation respecting image derivatives: a pde and scale-space based approach. Journal of Visual Communication and Image Representation 13 (1/2), 3–21.

Buehler, C., Bosse, M., McMillan, L., Gortler, S. J., Cohen, M. F., August 2001. Unstructured lumigraph rendering. In: Proceedings of ACM SIGGRAPH '01, Los Angeles. ACM Press, pp. 425–432.

Chai, J.-X., Tong, X., Chand, S.-C., Shum, H.-Y., July 2000. Plenoptic sampling. Proceedings of SIGGRAPH '00, New Orleans, 307–318.

Gortler, S., Grzeszczuk, R., Szeliski, R., Cohen, M. F., August 1996. The

lumigraph. In: Proceedings of SIGGRAPH '96, New Orleans. ACM Press, pp. 43–54.

Hartley, R., 1994. Euclidean reconstruction from uncalibrated views. In: Applications of Invariance in Computer Vision. Vol. 825 of Lecture Notes in Computer Science. Springer-Verlag, Berlin, pp. 237–256.

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

Heigl, B., Koch, R., Pollefeys, M., Denzler, J., Gool, L. V., September 1999. Plenoptic modeling and rendering from image sequences taken by a hand-held camera. In: Mustererkennung 1999. Springer-Verlag, Berlin, pp. 94–101.

Levoy, M., Hanrahan, P., August 1996. Light field rendering. In: Proceedings of SIGGRAPH '96, New Orleans. ACM Press, pp. 31–42.

Magnor, M., Ramanathan, P., Girod, B., November 2003. Multi-view coding for image-based rendering using 3-D scene geometry. IEEE Transactions on Circuits and Systems for Video Technology 13 (11), 1092–1106.

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

Schirmacher, H., Vogelgsang, C., Seidel, H.-P., Greiner, G., Nov. 2001. Efficient free form light field rendering. In: Workshop Vision, Modeling and Visualization, Saarbrücken, Germany. infix, Berlin, Amsterdam, pp. 249–256,528.

Tomasi, C., Kanade, T., April 1991. Detection and tracking of point features. Tech. rep., Carnegie Mellon University.

Tomasi, C., Kanade, T., November 1992. Shape and motion from image streams under orthography: A factorization method. International Journal of Computer Vision 9 (2), 137–154.

Triggs, B., June 1997. Autocalibration and the absolute quadric. In: Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society Press, pp. 609–614.

Vogt, F., Krüger, S., Schmidt, J., Paulus, D., Niemann, H., Hohenberger, W., Schick, C. H., 2004. Light fields for minimal invasive surgery using an endoscope positioning robot. Methods of Information in Medicine. To appear.

Zinßer, T., Gräßl, C., Niemann, H., August 2004. Efficient feature tracking for long video sequences. In: Pattern Recognition, Proceedings of 26th DAGM Symposium. Springer-Verlag, Berlin, to appear.