B. Heigl*, R. Koch‡, M. Pollefeys+, J. Denzler*, L. Van Gool+
heigl@informatik.uni-erlangen.de,
Reinhard.Koch@esat.kuleuven.ac.be:

Plenoptic Modeling and Rendering from Image Sequences taken by a Hand–Held Camera

Lehrstuhl für Mustererkennung, Universität Erlangen–Nürnberg, Germany*
Multimedia Systems, Institute of Computer Science, University of Kiel, Germany‡
ESAT–PSI, Katholieke Universiteit Leuven, Belgium+

Mustererkennung 1999 (DAGM)
Bonn, September 1999

Pages: 596–603

Abstract:
In this contribution we focus on plenoptic scene modeling and rendering from long image sequences taken with a hand–held camera. The image sequence is calibrated with a structure–from–motion approach that considers the special viewing geometry of plenoptic scenes. By applying a stereo matching technique, dense depth maps are recovered locally for each viewpoint.

View–dependent rendering can be accomplished by mapping all images onto a common plane of mean geometry and weighting them in dependence on the actual position of a virtual camera. To improve accuracy, approximating planes are defined locally in a hierarchical refinement process. Their pose is calculated from the local depth maps associated with each view without requiring a consistent global representation of scene geometry. Extensive experiments with ground truth data and hand–held sequences confirm performance and accuracy of our approach.

Keywords: Structure–from–Motion, Plenoptic Modeling

Plenoptic Modeling and Rendering from Image Sequences taken by a Hand–Held Camera

B. Heigl*, R. Koch‡, M. Pollefeys+, J. Denzler*, L. Van Gool+
heigl@informatik.uni-erlangen.de,
Reinhard.Koch@esat.kuleuven.ac.be

Lehrstuhl für Mustererkennung, Universität Erlangen–Nürnberg, Germany*
Multimedia Systems, Institute of Computer Science, University of Kiel, Germany‡
ESAT–PSI, Katholieke Universiteit Leuven, Belgium+

Abstract In this contribution we focus on plenoptic scene modeling and rendering from long image sequences taken with a hand–held camera. The image sequence is calibrated with a structure–from–motion approach that considers the special viewing geometry of plenoptic scenes. By applying a stereo matching technique, dense depth maps are recovered locally for each viewpoint.
View–dependent rendering can be accomplished by mapping all images onto a common plane of mean geometry and weighting them in dependence on the actual position of a virtual camera. To improve accuracy, approximating planes are defined locally in a hierarchical refinement process. Their pose is calculated from the local depth maps associated with each view without requiring a consistent global representation of scene geometry. Extensive experiments with ground truth data and hand–held sequences confirm performance and accuracy of our approach.
Keywords: Structure–from–Motion, Plenoptic Modeling

1 Introduction

In this contribution our goal is to create a model from a scene to render new views interactively. For this purpose two major concepts are known in literature. The first one is the geometry–based concept. The scene geometry is reconstructed from a stream of images and a single texture is synthesized which is mapped onto this geometry. For this approach, a limited set of camera views is sufficient, but specular effects can not be handled appropriately. The second major concept is image–based rendering. This approach models the scene as a collection of views all around the scene without an exact geometrical representation [9]. New (virtual) views are rendered from the recorded ones by interpolation in real–time. Approximative geometrical information optionally can be used to improve the results [4].

In this contribution we concentrate on the second approach. Up to now, the known scene representation has a fixed regular structure. If the source is an image stream taken with a hand–held camera, this regular structure has to be resampled. Our goal is to use the recorded images itself as scene representation

and to directly render new views from them. Geometrical information is considered as far as it is known and as detailed as the time for rendering allows. The approach is designed such, that the operations consist just of projective mappings which can efficiently be performed by the graphics hardware.

For each of these scene modeling techniques the camera parameters for the original views are supposed to be known. We retrieve them by applying known structure–from–motion techniques and adopting them to our special needs which result from the huge amount of images. Local depth maps are calculated applying stereo techniques on rectified image pairs.

Section 2 shows how to tackle this problem of camera calibration from images under special consideration of densely spaced view points. In section 3 we examine different methods for rendering new views from this calibrated sequence. We show how previous works compare to our approaches. In section 4 the experiments show the results of camera calibration and rendering. We compare rendering results of different methods.

2 Structure–From–Motion

To do a dense plenoptic modeling as described below, we need many views from a scene from many directions. For this, we can record an extended image sequence moving the camera in zigzag like manner. The camera can cross its own moving path several times or at least gets close to it. Known calibration methods usually only consider the neighborhoods within the image stream and use them for estimating the Fundmental matrix, see e.g. [2, 3, 5, 11]. No linking is done between views whose position is close to each other in 3–D space but which have a large distance in the sequence.

To deal with this problem, we therefore exploit the 2–D topology of the camera view points to further stabilize the calibration. We process not only the next sequential image but search for those images in the stream that are nearest in the topology to the current view point. This is done by estimating all fundamental matrices between image pairs and selecting the images with the least matching error. Typically we can establish a reliable matching to 3–4 neighboring images which improves the calibration considerably. The same approach was applied for geometric scene modeling in a second contribution of this conference [7] and details are described in [8].

In section 3 we will show how to use local depth maps for improving rendering results. Therefore dense correspondence maps are computed for adjacent image pairs of the sequence [6]. A disparity estimator based on dynamic programming is employed resulting in local depth maps [1].

3 Plenoptic Modeling and Rendering

We use the calibrated cameras to create a scene model for visualization. In [10] this is done by *plenoptic modeling*. The appearance of a scene is described

through all light rays (2–D) that are emitted from every 3–D scene point, generating a 5–D radiance function. Recently two equivalent realizations of the plenoptic function were proposed in form of the lightfield [9], and the lumigraph [4]. They handle the case when we observe an object surface within a transparent medium. Hence the plenoptic function is reduced to four dimensions. The radiance is represented as a function of light rays passing through the scene.

To create such a plenoptic model for real scenes, a large number of views is taken. These views can be considered as a collection of light rays with according color values. They are discrete samples of the plenoptic function. The light rays which are not represented have to be interpolated from recorded ones considering additional information on physical restrictions.

Often, real objects are supposed to be *Lambertian*, meaning that one point of the object has the same radiance value in all possible directions. This implies that two viewing rays have the same color value, if they intersect at a surface point. If specular effects occur, this is not true any more. Two viewing rays then have similar color values, if their direction is similar and if their point of intersection is near the real scene point which originates their color value. To render a new view we suppose to have a virtual camera looking into the scene. We determine those viewing rays which are *nearest* in the upper sense to those of this camera. The nearer a ray is to a given ray, the greater is its support to the color value.

3.1 Regular Grid Representations

The original 4–D lightfield [9] data structure employs a two–plane parameterization. Each light ray passes through two parallel planes with plane coordinates (s,t) and (u,v) (see figure 1). Thus the ray is uniquely described by the 4–tuple (u,v,s,t). The (s,t)–plane is the *viewpoint plane* in which all camera focal points are placed on regular grid points. The cameras are constructed such, that the (u,v)–plane is their common image plane and that their optical axes are perpendicular to it.

From the two–plane parameterization new views can be rendered by placing a virtual camera on an arbitrary viewing position with arbitrary parameters (e.g. focal length) and intersecting each viewing ray with the two planes at (s,t,u,v). The resulting radiance is a look–up into the regular grid. For rays passing in between the (s,t) and (u,v) grid coordinates an interpolation is applied that will degrade the rendering quality depending on the scene geometry. In fact, the lightfield contains an implicit geometrical assumption: The scene geometry is planar and coincides with the focal plane (figure 2). Deviation of the scene geometry from the focal plane causes image warping. Figure 2 shows that the radiance of the viewing ray r is interpolated from radiance values l_{i-1} and l_i of neighboring camera viewpoints, depending on the geometrical deviation from the focal plane.

Linear interpolation between the viewpoints in (s,t) and (u,v) introduces a blurred image with ghosting artifacts. In reality we will always have to choose

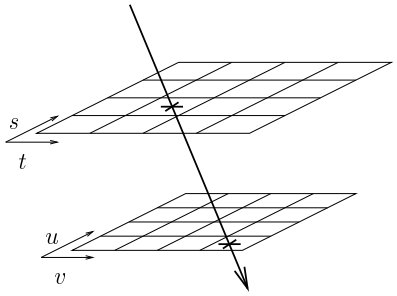

Figure1. 4–D parameterization of viewing rays in the regular grid representation of the lightfield.

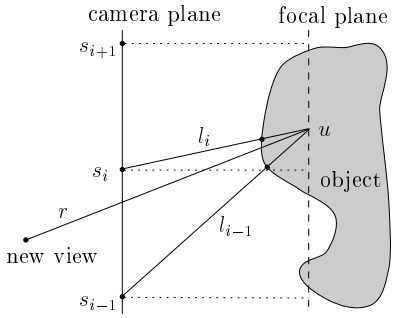

Figure2. Viewpoint interpolation between s_i and s_{i-1}.

between high density of stored viewing rays with high data volume and high fidelity, or low density with poor image quality.

If we have 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. In [4] a method is shown for resampling this regular two plane parameterization from real images recorded from arbitrary positions (*rebinning*). The required regular structure is resampled and gaps are filled by applying a multi–resolution approach, considering depth corrections. The disadvantage of this *rebinning* step is that the interpolated regular structure already contains inconsistencies and ghosting artifacts because of errors in the scantily approximated geometry. To render views, a depth corrected look–up is performed. During this step, the effect of ghosting artifacts is repeated, so duplicate ghosting effects occur.

3.2 Representation with Recorded Images

Our goal is to overcome these problems described in the last section by relaxing the restrictions imposed by the regular lightfield structure and to render views directly from the calibrated sequence of recorded images with use of local depth maps. Without loosing performance we directly map the original images onto one or more planes viewed by a virtual camera.

2–D Mapping. The following approaches will use this formalism to map images onto planes and vice versa. We define a local coordinate system on A giving one point $\boldsymbol{a}_0$ on the plane and two vectors $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ spanning the plane. So each point $\boldsymbol{p}$ of the plane can be described by the coordinates x_A, y_A: $\boldsymbol{p} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_0)\,(x_A, y_A, 1)^{\mathrm{T}}$. The point $\boldsymbol{p}$ is perspectively projected into a camera which is represented by the 3×3 matrix $\boldsymbol{M} = \boldsymbol{K}\boldsymbol{R}^{\mathrm{T}}$ and the projection center $\boldsymbol{c}$. Matrix $\boldsymbol{R}$ is the orthonormal rotation matrix and $\boldsymbol{K}$ is an upper triangular calibration matrix. The resulting image coordinates x, y are determined

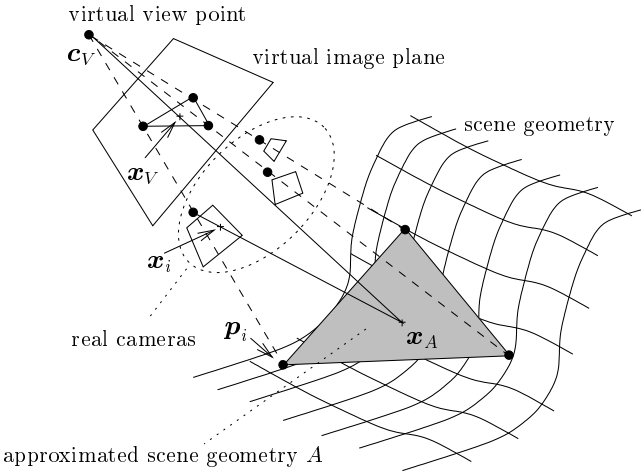

Figure3. Drawing triangles of neighboring projected camera centers and approximating scene geometry by one plane for the whole scene, for one camera triple or by several planes for one camera triple.

by $\rho(x, y, 1)^{\mathrm{T}} = \boldsymbol{M}\boldsymbol{p} - \boldsymbol{M}\boldsymbol{c}$. Inserting upper equation for $\boldsymbol{p}$ results in

$$\rho \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \boldsymbol{M}(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_0 - \boldsymbol{c}) \begin{pmatrix} x_A \\ y_A \\ 1 \end{pmatrix}.$$

The value ρ is an unknown scale factor. Each mapping between a local plane coordinate system and a camera can be described by a single 3×3 matrix $\boldsymbol{B} = \boldsymbol{M}(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_0 - \boldsymbol{c})$.

Mapping via global plane. In a first approach, we approximate the scene geometry by a single plane A by minimizing the least square error. We map all given camera images onto A and view it plane through a virtual camera. This can be achieved by directly mapping the coordinates x_i, y_i of image i into the virtual camera coordinates $(x_V, y_V, 1)^{\mathrm{T}} = \boldsymbol{B}_V \boldsymbol{B}_i^{-1}(x_i, y_i, 1)^{\mathrm{T}}$. Therefore, we can perform a direct look–up into the originally recorded images and determine the radiance by interpolating the recorded neighboring pixel values. This technique is similar to the lightfield approach [9] which implicitly assumes the uv–plane as the plane of geometry.

Therefore, to construct a specific view, we have to interpolate between neighboring views. Those views give the most support to the color value of a particular pixel whose projection center is close to the viewing ray of this pixel. This is equivalent to the fact that those views give the most support to a specified pixel whose projected camera centers are close to its image coordinate. We restrict the support to the nearest three cameras (see figure 3). 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. Each triangle is drawn as a sum of three triangles. For each camera we look up the color values in the original image like described above and multiply them with weight 1 at the corresponding corner and with weight 0 at both other like. In between, the weights are interpolated linearly similar to Gouraud–Shading. Within the triangle, the sum of weights is 1 at each point. The total image is built as a mosaic of these triangles. Although this technique assumes a very sparse approximation of geometry, the rendering results just show small ghosting artifacts (see section 4).

Mapping via local planes. 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. This improves the accuracy further 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. The depth values are given as functions z_i of the coordinates in the recorded images $z_i((x_i, y_i, 1)^{\mathrm{T}})$. They describe the distance of a point perpendicular to the image plane. Using this depth function, we calculate the 3–D coordinates of those scene points which have the same 2–D image coorinates in the virtual view as the projected camera centers of the real views. The 3–D point $\boldsymbol{p}_i$ which corresponds to the real camera i can be calculated as $\boldsymbol{p}_i = z_i(\boldsymbol{M}_i\boldsymbol{d}_i)\boldsymbol{d}_i + \boldsymbol{c}_i$, where $\boldsymbol{d}_i = n(\boldsymbol{c}_i - \boldsymbol{c}_V)$. The function n scales the given 3–D vector such, that its third component equals one. We can interpret the points $\boldsymbol{p}_i$ as the intersection of the line $\overline{\boldsymbol{c}_V\boldsymbol{c}_i}$ with the scene geometry. Knowing the 3–D coordinates of triangle corners, we can define a plane through them and apply the same rendering technique as described above.

Refinement. Finally, if the triangles exceed a given size, they can be subdivided into four sub–triangles by splitting the three sides into two parts, each. For each of these sub–triangles, a separate approximative plane is calculated in the above manner. We determine the midpoint of the side and use the same look–up method as used for radiance values to find the corresponding depth. After that, we reconstruct the 3–D point and project it into the virtual camera resulting in a point near the side of the triangle. Of course, further subdivision can be done in the same manner to improve accuracy. Especially, if just few triangles contribute to a single virtual view, this subdivision is really necessary. It should be done in a resolution according to performance demands and to the complexity of geometry.

4 Experiments

We have shown how to calibrate an image sequence which is taken with a hand–held camera, and how to use these calibrated images to render new virtual views from the scene. In this section, experiments show the differences and properties of the approaches.

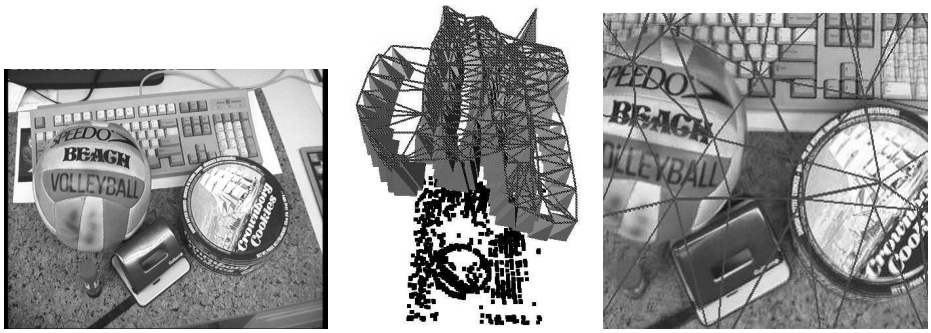

Figure4. Left: one image of the original sequence. Middle: Calibration result. Cameras are shown as pyramids and their topological mesh is drawn with lines. Right: Reconstructed scene view using one plane per image triple.

Figure5. Details of rendered images showing the differences between the approaches: one global plane of geometry (left), one local plane for each image triple (middle) and refinement of local planes (right).

We have tested our approaches with an uncalibrated sequence of 187 images showing an office scene. Figure 4 (left) shows one particular image. A digital consumer video camera was swept freely over a cluttered scene on a desk, covering a viewing surface of about 1 m^2. Figure 4 (middle) shows the calibration result. In another experiment with ground–truth data we improved the calibration accuracy of 2.31% of the mean object distance to 1.41% by extending the standard structure–from–motion technique by scanning the viewpoint surface as described in section 2. A detailed discussion of the reconstruction accuracy can be found in [8].

One result of a reconstructed view is shown in figure 4 (right). Figure 5 shows details for the different methods. In the case of one global plane (left image), the reconstruction is sharp where the approximating plane intersects the actual scene geometry. The reconstruction is blurred where the scene geometry diverges from this plane. In the case of local planes (middle image), at the corners of the triangles, the reconstruction is almost sharp, because there the scene geometry is considered directly. Within a triangle, ghosting artifacts occur where

the scene geometry diverges from the particular local plane. If these triangles are subdivided (right image) these artifacts are reduced further.

5 Further Work and Conclusions

In this contribution, we have shown how to use images taken by a hand–held camera which is moved around a scene to directly render views *without* constructing a regular grid representation for the plenoptic function as it is known from literature. The quality of rendered images can be varied by adjusting the resolution of the considered scene geometry.

Up to now, our approaches are calculated in software. But they are designed such, that using alpha blending and texture mapping facilities of graphics hardware, rendering will be done in real–time.

Acknowledgments. This work was partially funded by the German Research Foundation (DFG) under the grant number SFB 603. We also acknowledge financial support from the Belgian project IUAP 04/24 "IMechS".

References

1. L. Falkenhagen. Hierarchical block-based disparity estimation considering neighborhood constraints. In *Intern. Workshop on SNHC and 3D Imaging*, Rhodes, Greece, 1997.
2. O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. In *Proceedings ECCV '92*, pages 563–578. Springer, 1992.
3. O. Faugeras, Q.-T. Luong, and S. Maybank. Camera self-calibration - theory and experiments. In *Proceedings ECCV '92*, pages 321–334. Springer, 1992.
4. S. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proceedings SIGGRAPH '96*, pages 43–54. ACM Press, 1996.
5. R. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *Proceedings ECCV '92*, pages 579–587, 1992.
6. R. Koch, M. Pollefeys, and L. Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *Proceedings ECCV '98*, 1998.
7. R. Koch, M. Pollefeys, and L. Van Gool. Robust calibration and 3d geometric modeling from large collections of uncalibrated images. In *Mustererkennung '99*, 1999.
8. R. Koch, M. Pollefeys, B. Heigl, L. Van Gool, and H. Niemann. Calibration of hand-held camera sequences for plenoptic modeling. In *Proceedings ICCV '99*, 1999. to appear.
9. M. Levoy and P. Hanrahan. Lightfield rendering. In *Proceedings SIGGRAPH '96*, pages 31–42. ACM Press, 1996.
10. L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. In *Proceedings SIGGRAPH '95*, pages 39–46, 1995.
11. M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proceedings ICCV '98*, Bombay, India, 1998.