Ingo Scholz, Joachim Denzler, Heinrich Niemann

Calibration of Real Scenes for the Reconstruction of Dynamic Light Fields

appeared in:
IAPR Workshop on Machine Vision Applications 2002 (MVA 2002)
Nara, Japan
pp. 32–35

Calibration of Real Scenes for the Reconstruction of Dynamic Light Fields

Ingo Scholz [*], Joachim Denzler and Heinrich Niemann
Universität Erlangen–Nürnberg
Lehrstuhl für Mustererkennung (Informatik 5)
Martensstr. 3, 91058 Erlangen, Germany
E-mail: {scholz,denzler,niemann}@informatik.uni-erlangen.de
www: http://www5.informatik.uni-erlangen.de

Abstract

The classic light field and lumigraph are two well-known approaches to image-based rendering, and subsequently many new rendering techniques and representations have been proposed based on them. Nevertheless the main limitation remains that in almost all of them only static scenes are considered. In this contribution we describe a method for calibrating a scene which includes moving or deforming objects from multiple image sequences taken with a handheld camera. For each image sequence the scene is assumed to be static, which allows the reconstruction of a conventional static light field. The dynamic light field is thus composed of multiple static light fields, each of which describes the state of the scene at a certain point in time. This allows not only the modeling of rigid moving objects, but any kind of motion including deformations.

In order to facilitate the automatic calibration, some assumptions are made for the scene and input data, such as that the image sequences for each respective time step share one common camera pose and that only the minor part of the scene is actually in motion.

1 Introduction

In recent years the field of image-based rendering has become a very popular research topic. The light field [4] and the lumigraph [1] are two similar and often used approaches for modeling objects or scenes from a set of input images and without prior knowledge of scene geometry. One of their advantages over conventional model-based rendering techniques is that they allow photo-realistic rendering of real scenes or objects, while computation time is independent of the complexity of scene geometry.

While it is already possible to generate light fields from real but static scenes and render high-quality images from them [3], these light fields are not applicable to dynamic scenes, i.e. scenes that vary over time. Nevertheless a lot of applications can be thought of where dynamic light fields would be useful. In endoscopic surgery [10] for instance an automatically generated light field would allow the physician to view the organ he is operating from any view point without having to move the endoscope, thus reducing the strain on the patient. The problem is that many human intestines are permanently in motion so that the static light field is insufficient to model them.

We currently focus our research on solutions for applications like the above, which can be generally described as real scenes containing periodically moving objects. At present we also require the scene to have a static background, while the dynamic part of the scene is smaller than the rest. The extension of static light fields to dynamic scenes and objects gives rise to several problems:

- Calibration of scenes including moving objects has to cope with unreliable point correspondences, requiring the identification of different time frames and the distinction of static and dynamic parts of the scene.
- The amount of data to be stored is already large for static light fields, but dynamic light fields further increase the dimension of the parameter space.
- Extended rendering techniques are required for rendering images at arbitrary points in time.

In this contribution we will concentrate mostly on the first of these problems, the calibration of dynamic scenes. Many applications, like the one described above, do not allow the use of calibration patterns in the scene [1] or the placement of the camera at specific known positions in the world reference frame [4]. Therefore we pursue the approach described in [3], which is to calibrate the cameras automatically from the input image sequence using structure from motion algorithms. The required point correspondences are created by automatically extracting and tracking point features in the scene.

The main problem in automatically calibrating dynamic scenes is that it cannot be determined whether the movement of a point feature from image to image is due to the movement of the camera or of an object in the scene. For being able to use the latter points for calibration, the deformation of the scene itself would have to be known very precisely. Unfortunately this knowledge cannot be gained if the camera movement is unknown. In order to break out of this vicious circle we assume that for each time step an image sequence is available, which is then again of a static scene. The dynamic light field is thus composed of multiple static light fields, one for each time step.

The modeling of dynamic scenes and objects is currently a very active topic of research. Solutions have been proposed for handling multiple rigid moving objects in a scene [6] or modeling non-rigid objects [8]. While good results are already reached here, these approaches still need to constrain the underlying projection models or the type of object movement. In our approach on the other hand we can rely

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

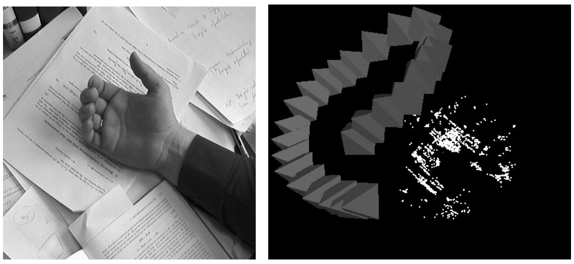

Figure 1: Left: First image of an example image sequence. Right: Calibration results for this image sequence. Each pyramid represents a camera, the dots are the 3D points on the scene surface.

on the relative robustness and quality of established methods for calibrating static scenes, while the modeling of dynamics is done through the combination of the results.

Thus, the main problem we have to address is how to combine these static light fields and transform them into a common coordinate system. This article will concentrate on this registration task, while the rendering of arbitrary views from the resulting light field will only be addressed briefly.

The registration step will be described in detail in the next section, followed by a section treating the process of image rendering. Experimental results will be given in Section 4, and Section 5 offers some concluding remarks and an outlook to future work.

2 Calibrating Dynamic Light Fields

Instead of putting together a dynamic scene from one static light field for each rigid but moving object, as it was described in [5], we subdivide the dynamic scene into k time steps and model each with a complete static light field of the scene. We are thus able to not only model rigid but also deformable objects in the scene.

The input images we will use in the following to reconstruct a dynamic light field need to fulfill two main requirements. First, one image sequence must be available for each time step so that the k static light fields can be reconstructed from them. Second, for two consecutive image sequences the last camera of the first sequence must have the same pose as the first camera of the second sequence, which means that the two sequences have one camera in common. The dynamic light field is reconstructed from this input data by first calibrating the individual image sequences and then registering the resulting threads of camera positions with each other. Finally a refinement step can be applied which calibrates all cameras together. The assumption which must be made for this last step to work is that dynamic objects only influence the lesser part of the visible scene. The three calibration steps will be described in the following.

2.1 Static Light Field Calibration

Each image sequence is calibrated independently following the approach of [2] which involves a three-step process. The first step is the establishment of point correspondences between the images through feature tracking. In the second step an initial subsequence is calibrated using factorization methods, and in the last step the remaining cameras are added by using the reconstructed 3-D points as a calibration pattern.

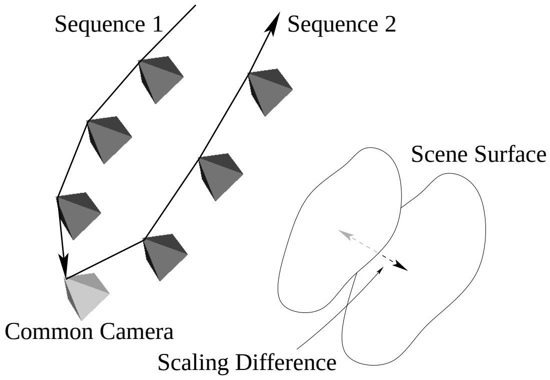

Figure 2: Registration of two image sequences. The incorrect scaling results in different distances of the scene surface point clouds from the cameras.

Apart from the projection matrix of each camera the calibration also yields a set of 3-D points which correspond to the 2-D feature points used for calibration. The cameras and 3-D point sets for the calibration of an example image sequences are shown in Figure 1. The coordinate systems of the reconstructed cameras of two image sequences now differ from each other by a rotation, a translation and an unknown scale factor, and need to be registered with each other in the following steps.

2.2 Registration

The rotation and translation can be determined by the fact that two cameras of a pair of consecutive image sequences have the same pose. The transformation is done by first mapping one of the two cameras into the origin of its coordinate system and then to the pose of the other camera. The same transformation is then applied to all remaining cameras of this sequence.

Figure 2 depicts such a registration of two image sequences using a common camera. It also shows the effect of the missing scaling step, which results in different distances of the two 3-D point clouds from the camera positions.

The scale factor is obtained by considering the center of mass of the 3-D points in each image sequence. As the sequences were taken of the same scene, the centers of mass are assumed to be roughly at the same position. Furthermore, we require that in two consecutive image sequences similar features are tracked on the dynamic objects, or that their movement is small. The scale factor is computed as the ratio of the distances of the centers of mass from the two equal cameras of two consecutive sequences. The cameras and 3-D points of the respective second sequence are then scaled with this factor.

2.3 Refinement

Now that the cameras of all light fields are transformed to the same coordinate system, a further refinement of the calibration can be performed. The camera positions form a 3-D mesh in which neighbours with similar views on the scene can easily be identified. In order to make sure that the corresponding images show similar parts of the scene the viewing direction of two neighbouring cameras must be similar, too. Values for the allowed distance and viewing direction difference are calculated as multiples of the average values for all pairs of subsequent cameras.

Using these neighbourhoods a second tracking step is invoked. This time, no new features are added but only those are tracked further that were used for the first calibration. The calibration process removes outliers and features that could only be tracked over 2 or 3 frames, leaving only the more robust ones.

Tracking is performed in a two-step loop for each image sequence:

1. The existing features are tracked to the other image sequences following the neighbourhood links established before.

2. These additional features are now propagated through the other image sequences. Depending on the effort to be spent this can be just the preceding and the following image sequence of the current one, or all other image sequences. The complexity in the latter case of course increases quadratically with the number of time steps.

Through this obviously time-consuming process, the formerly mostly unrelated sequences – except for the one frame with each neighbour – are now linked together over these new feature correspondences.

The disadvantage of these correspondences is that any of them could be positioned on a moving, i. e. dynamic, part of the scene. These *dynamic features* can be considered equivalent to erroneously tracked points, and could severely perturb the calibration process. Nevertheless, by postulating that only a minor part of the scene is actually in motion the calibration algorithm proved to be robust enough to handle these outliers.

The reason is that after the factorization of a subset of only a few cameras (see Section 2.1) the calibration gets extended to the rest of the sequences by classical calibration techniques [9]. In this step the 3-D points acquired through factorization are used as calibration pattern, which is extended after calibration of each new image by triangulating the features found in it. At this point 3-D points with a high back-projection error are discarded, which is often the case for dynamic features from two different image sequences.

On the other hand, features on a dynamic part of the scene are unproblematic as long as they are not tracked to another image sequence at a different time step.

3 Rendering

Since we are using a hand-held camera for capturing the images for our dynamic light fields, the camera positions may be distributed almost arbitrarily in space. Therefore the most obvious parameterization is that of a free form light field [3]. Other parameterizations, like using two planes [1], would require a preceding warping step which decreases image quality.

For rendering images from a dynamic light field we extended an already existing hardware-accelerated free form renderer [7] to handle an arbitrary number of static light fields. The difference to rendering images from static light fields is that a timestamp is now required as an additional parameter. Rendering images at known time steps, i. e. those where the image sequences were taken, can thus be done without additional effort.

Generating views of the scene at arbitrary positions in time on the other hand is a much more difficult problem, and many different solutions can be thought of. One approach is to first render the views for the earlier and later integer time steps and then generate images at any intermediate time by interpolation.

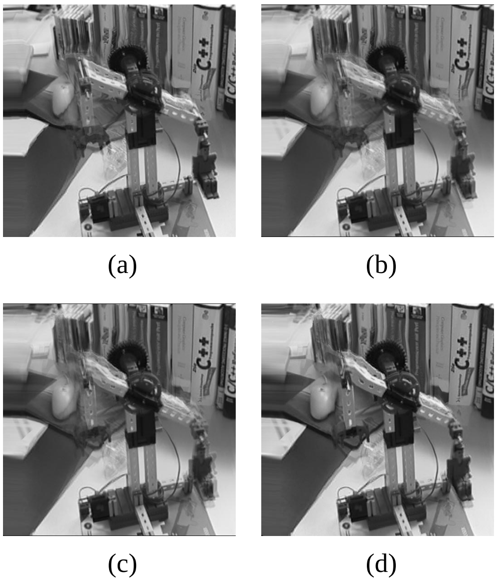

Figure 3: Light field of a toy rotor. Images (a) and (d) are of integer time steps, (b) and (c) intermediate steps generated by cross-fading.

Since the emphasis of our current work is not on the rendering of light fields but on their generation, we only implemented the basic technique of creating new views by cross-fading the two available images, weighted by their distance to the desired timestamp. The result can be seen in Figure 3, where images (a) and (d) are of two subsequent integer time steps, and images (b) and (c) the two steps in between.

Using additional information about the scene, which is already available through the calibration process, more sophisticated methods can be applied as well. The back-projections of the known 3-D points into the rendered image can be used as control points for the application of different kinds of image warping techniques which are widely used in computer graphics [11].

4 Experiments

The experiments described in the following section were conducted to analyze the quality of registration of the image sequences. For this purpose three dynamic light fields were created as described in Section 2. Each of them shows a different dynamic object in front of a static background. Rendered example images of each scene are shown in Figure 4, and column 2 of Table 1 states the number of available time steps for each scene.

A measure for the quality of image sequence registration is the shift of the background for two rendered images of different time steps, but as seen with exactly the same camera pose. Putting this shift into numbers is difficult, and we chose the average absolute pixel difference as a measure. While this lacks some quantitative expressiveness, it can still give a good qualitative impression.

Since only the background shift was to be considered, the dynamic objects in the foreground were removed by hand-coloring them in black. These and any other colorless parts of the images were ignored in the difference. Columns 5 and 6 of Table 1 show the pixel differences for the simple image sequence concatenation of Section 2.2 compared to the value after the refinement described in Section 2.3. In all

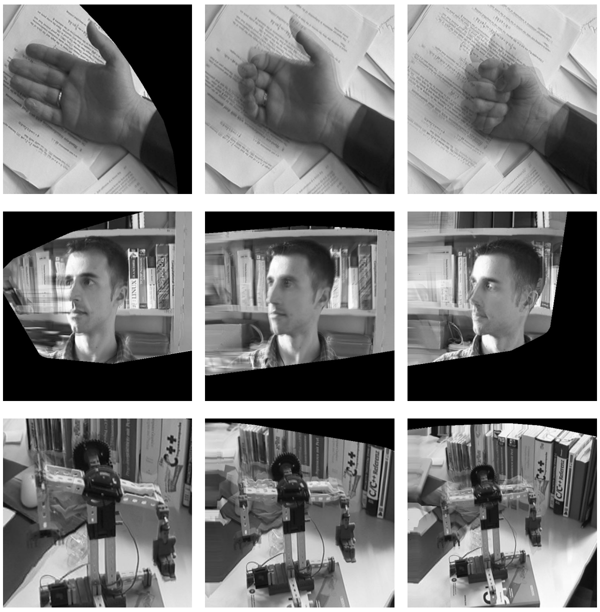

Figure 4: Example images from the scenes used for the experiments. The images in the first two rows were rendered using a constant camera pose, while for the third row the camera was moved and zoomed simultaneously.

		# Image diffs		Mean diff	
Scene	# Seq.	concat	refine	concat	refine
Hand	5	10	10	17.3	12.0
Head	6	18	15	30.8	17.6
Rotor	8	30	28	40.3	12.7

Table 1: Comparison of background shifts in the example scenes using mean pixel difference.

scenes the refinement step clearly improves the registration, which can be seen in the example in Figure 5.

In order to ensure the validity of this comparison the test images were always rendered with approximately the same visible object sizes. An exact match was not possible since the coordinate systems differ before and after refinement. The values in columns 3 and 4 of Table 1 denote the number of image comparisons performed. These may vary since not all camera poses generate valid images for every time step, depending on the distance to reference images.

5 Conclusion and Future Work

In the preceding sections we described a method for reconstructing a dynamic light field from multiple image sequences, each referring to a different time step. The preconditions are that image sequences of consecutive time steps share a common frame concerning the camera pose and parameters, and that only the lesser part of the scene is in motion. By calibrating each image sequence independently and concatenating the results, a good registration of the camera sequences can be achieved. A subsequent refinement step can further improve the quality of registration.

By storing each time step as an individual light field the modeling of arbitrary movements of the scene is possible. Images at intermediate time steps can be rendered by crossfading images from neighbouring known light fields.

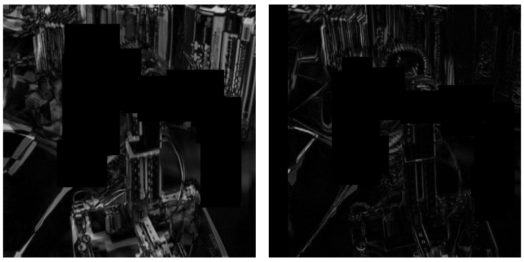

Figure 5: Difference images for time steps 4 and 7 of the Rotor sequence from similar camera positions before and after refinement.

Our future research will focus on the relaxation of the above requirements, so that the object can be in motion while being recorded. This could be achieved by regarding periodic movements and determining their frequency. The medical applications mentioned in the introduction could thus be handled.

In addition to that the two issues of rendering images at any point in time and of efficiently storing dynamic light fields have to be addressed as well.

References

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

[2] B. Heigl. *Plenoptic Scene Modeling from Uncalibrated Image Sequences*. 2002. Dissertation, to appear.

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

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

[5] W. Li, Q. Ke, X. Huang, and N. Zheng. Light field rendering of dynamic scene. *Machine Graphics and Vision*, 7(3), 1998.

[6] R. Manning and C. Dyer. Affine calibration from moving objects. In *Proceedings of the 8th IEEE International Conference on Computer Vision*, volume I, pages 494–500, 2001.

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

[8] L. Torresani, D. B. Yang, E. J. Alexander, and C. Bregler. Tracking and modeling non-rigid objects with rank constraints. In *Proceedings of IEEE Conference Computer Vision and Pattern Recognition*, 2001.

[9] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Addison–Wesley, Massachusets, 1998.

[10] F. Vogt, D. Paulus, and H. Niemann. Highlight Substitution in Light Fields. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 637–640, Rochester, USA, September 2002. IEEE Computer Society Press.

[11] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, 1990.