3D Reconstruction and Rendering from Image Sequences

R. Koch, J.-F. Evers-Senne, J.-M. Frahm, K. Koeser
Institute of Computer Science and Applied Mathematics
Christian-Albrechts-University of Kiel, 24098 Kiel, Germany
email: rk@informatik.uni-kiel.de

Abstract

This contribution describes a system for 3D surface reconstruction and novel view synthesis from image streams of an unknown but static scene. The system operates fully automatic and estimates camera pose and 3D scene geometry using Structure-from-Motion and dense multi-camera stereo reconstruction. From these estimates, novel views of the scene can be rendered at interactive rates.

1 INTRODUCTION

Image Based Rendering (IBR) is an active research field in the computer vision and graphics community. In the last decade, many systems were proposed that synthesize novel views based on collections of real views of a scene. Those systems differ with respect to the amount of interactivity for novel view selection, the ability to compensate scene depth effects (parallax) etc. For a recent review on IBR techniques see [8].

In this contribution, we will describe a system for 3D surface reconstruction and novel view synthesis from images of an unknown static scene [4]. In a first step, a structure from motion (SfM) approach is employed to estimate the unknown camera poses and intrinsic calibration parameters of the camera throughout the sequence [11]. The images may come either from a set of closely spaced photographic still images, a hand-held video camcorder, or a multi-camera rig with rigidly coupled digital firewire cameras. Together with the calibration, a sparse set of 3D feature points is estimated based on the static scene hypothesis. Those features already contain a sparse 3D scene description. To refine the 3D scene geometry, dense depth maps are estimated from the now calibrated input images in a second step [10]. Thus, for each recorded image, the associated calibration, 3D pose, and a dense depth map is stored [9].

These data can be used in many ways. One way would be to reconstruct a consistent 3D scene surface from all views by triangulating a 3D wireframe surface from all depth maps. The surface can then be textured with the real images, forming a view-dependent texture map surface (VDTM) [1]. For this, the topology problem must be solved to distinguish between connected and occluded surfaces from all views. Another way would be to render directly from the real views using depth-compensated warping [2]. In this case, local surface geometry between adjacent views is sufficient. In our contribution we describe ways to render interpolated views using view-dependent geometry and texture models (VDGT) [3].

We will describe the system and its components in the following section, followed by some experiments and evaluation of the approach.

2 SYSTEM OVERVIEW

Figure 1 gives an overview on the components of the system. The complete system can be divided into an offline data acquisition and an online rendering part. In the offline part, the images are preprocessed to estimate calibration and depth maps for each view. In the online rendering, the given data set is used to render novel views at interactive rates.

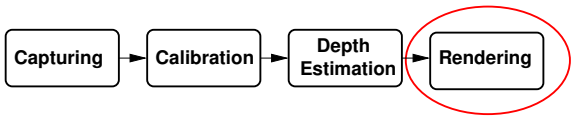

Figure 1: Block diagram of the reconstruction and rendering system.

2.1 Offline Data Acquisition

Relative camera calibration is obtained with a structure from motion approach similar to [11]. A 2D feature detector (Harris Detector [6] or structure tensor [5]) extracts dominant 2D intensity corners in the images. To find correspondence matches between different views of the scene, we track the features throughout the 2D sequence using the KLT-tracker [12] or with correlation-based corner matching. The correspondence matching is guided by the epipolar constraint and robust matching statistics using random

Figure 2: Images of the original scene.

sampling consensus (RANSAC) [7].

The image correspondences are tracked through many images viewing the same scene. All 2D correspondences result from projection of the same 3D feature into the images, hence we compute the 3D intersection of all viewing rays by simultaneous estimation of the camera poses and 3D feature points. Bundle adjustment will compute the best possible fit and give the relative camera pose of all views and the corresponding 3D features. We cannot compute absolute camera pose as the overall scale of the scene is not known, but a scaled metric pose estimate is determined [7]. Figure 2 shows some images of the scene used for tracking and reconstruction. Figure 4 shows an overview image of the scene and the resulting camera pose and 3D feature estimates after SfM tracking.

Figure 3: Dense depth maps of scene, depth color coded (dark=near, light=far, black=undefined).

After camera calibration, a dense and robust depth estimate is needed for every pixel of the different views. This can be achieved by multi-view stereoscopic depth estimation [10, 13]. For each view, all spatially neighboring camera views are used to estimate per-pixel depth, which is stored in a depth map. Results of this depth estimate can be seen in figure 3. The depth maps are usually dense and relative depth error is around 1% relative depth deviation. These data form the basis to the 3D reconstruction and online IBR generation.

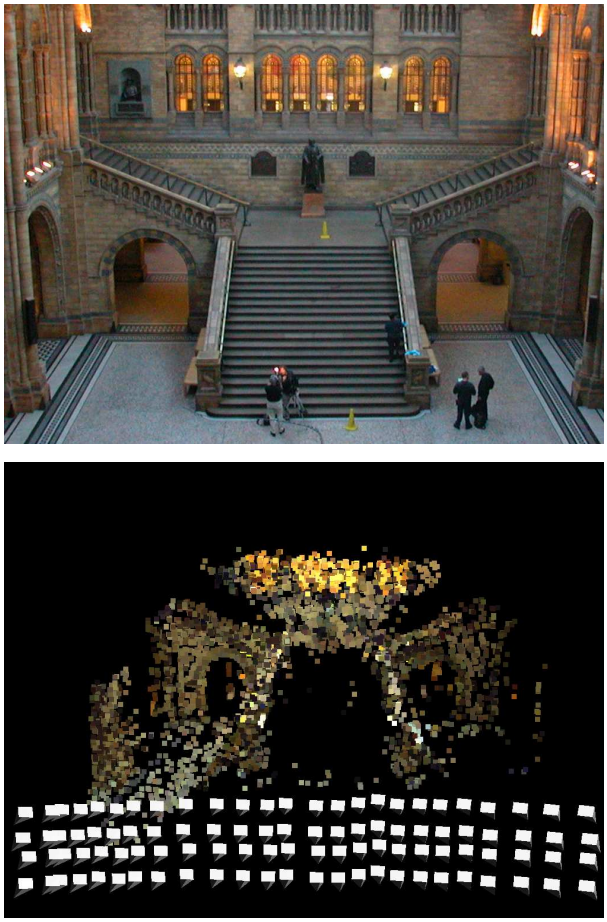

Figure 4: Overview image (top) and 3D calibration and tracks of scene (bottom). The little pyramids show the camera positions, the colored points give the 3D feature positions of salient tracked features.

2.2 Interactive Online Rendering

The calibrated views and the preprocessed depth maps are used as input to the image-based interactive rendering engine. The user controls a virtual camera which views the scene from novel viewpoints. The novel view is interpolated from the set of real calibrated camera images and their associated depth maps. During rendering it must be decided which camera images are best suited to interpolate the novel view, how to compensate for depth changes and how to blend the texture from the different images. For large and complex scenes hundreds or even thousands of images have to be processed. All these operations must be performed at interactive frame rates of 10 fps or more. We address these issues in the following section:

- Selection of best real camera views,
- fusion of multiview geometry from the views,

- viewpoint-adaptive mesh generation,
- viewpoint-adaptive texture blending.

For each novel view to be rendered, the most suitable real views must be selected for interpolation. The cameras are ranked based on similarity in view point, viewing direction, and common field of view with the novel view. For further detail we refer to [1].

Rendering with View dependent Geometry and Texture (VDGT): The ranked cameras and their associated depth samples are now used to interpolate novel views. Since the novel view may cover a field of view that is larger than any real camera view, we have to fuse views from different cameras into one locally consistent image. Efficient hardware-accelerated image warping is employed to map the different real views into the novel viewpoint. Therefore, we generate a warping surface from a regular grid that is placed in the image plane of the virtual camera. The warping surface will have to be updated for each camera motion at interactive rates. Therefore, warping uses a scalable coarse geometric approximation of the real depth maps. Using this approximation as a coarse 3D surface model, the novel view is rendered by blending the rendered coarse models into one consistent novel view. For texture blending, one may decide to either select the best-ranked camera (single-texture mode) or to blend all associated camera textures on the surface (multi-texture mode). Proper blending of all textures will result in smoother transition between views but with higher rendering costs for multi-pass rendering.

Rendering from Multiple Local Models (MLM): Instead of the backward warping used for VDGT, one may use a forward mapping by computing a set of individual local models. These models could be a direct meshing of the depth map for each view or a set of depth layers [2]. In that case, the rendering is simplified to texture mapping of all local models and rendering them into the new view. This is fully hardware-accelerated and can be performed very fast. The drawback is that no consistency between the different models is guaranteed and that holes may remain in the rendered view.

3 RESULTS

In this section we will discuss modeling and rendering results with the proposed methods.

The depth maps obtained from the modeling are not dense, as can be seen by the black regions in figure 3. This will cause the depth compensated warping to fail, unless the holes are interpolated properly. Figure 5 shows rendering results by view interpolation from adjacent camera views. The top image shows rendering results with the four best ranked cameras and depth interpolation, using VDGT. It can be seen that the wrong depth range in the back of the archway leads to blending artifacts, seen as a slight blur due to inconsistent depth in the different cameras. However, the impression of the image is quite smooth and the artefacts are not very visible. In the bottom image, rendering with MLM gives a sharper impression as the depth buffer selects the nearest model surface only, but due to inconsistencies, the image appears inconsistent in some parts and some holes remain.

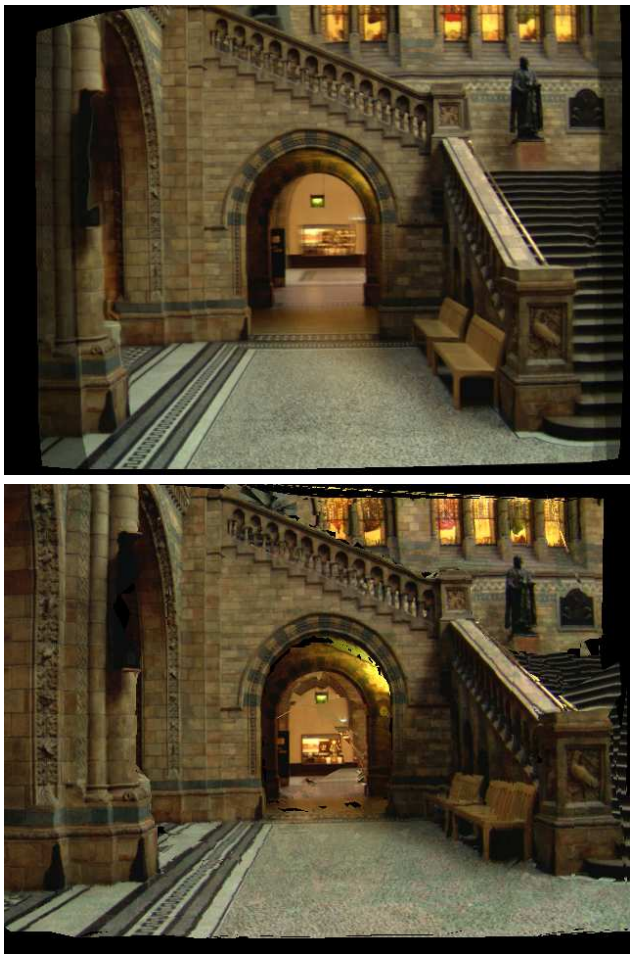

Figure 5: IBR results for view rendering using VDGT (top) and MLM (bottom).

4 CONCLUSIONS

We have presented a system for the automatic IBR from uncalibrated, handheld image sequences. The camera path was calibrated and nearly dense depth maps were computed, leaving a set of calibrated and textured depth maps for depth-compensated interpolation.

Two different methods for view interpolation were discussed. The rendering results show that the generated quality is still not sufficient for seamless rendering from arbitrary extrapolated image positions. One issue to investigate further is the proper handling of unmodeled depth regions and the problem of global seamless integration of local models.

Acknowledgments

This work was funded partially by the European projects IST 2000-28436 ORIGAMI and IST-2003-2013 MATRIS.

References

[1] J.-F. Evers-Senne and R. Koch. Image based interactive rendering with view dependent geometry. In *Eurographics 2003*, Computer Graphics Forum. Eurographics Association, 2003.

[2] J.-F. Evers-Senne and R. Koch. Image based rendering from handheld cameras using quad primitives. In *Vision, Modeling, and Visualization VMV: proceedings*, Nov. 2003.

[3] J.-F. Evers-Senne and R. Koch. Interactive rendering with view-dependent geometry and texture. In *Sketches and Applications SIGGRAPH 2003*, 2003.

[4] J.-F. Evers-Senne, J. Woetzel, and R. Koch. Modelling and rendering of complex scenes with a multi-camera rig. In *1st European Conference on Visual Media Production (CVMP 2004), London, United Kingdom*, March 2004.

[5] W. Förstner. A feature based correspondence algorithm for image matching. In *International Archives of Photogrammetry and Remote Sensing*, volume 26-3/3, pages 150–166. Rovaniemi, 1986.

[6] C. Harris and M. Stephens. A combined corner and edge detector. In *4th Alvey Vision Conference, Manchester*, pages 147–151, 1988.

[7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge university press, 2000.

[8] R. Koch and J.-F. Evers-Senne. View synthesis and rendering methods. In *3D Video Communications*. Wiley, 2005.

[9] R. Koch, J. Frahm, J.-F. Evers-Senne, and J. Woetzel. Plenoptic modeling of 3d scenes with a sensor-augmented multi-camera rig. In *Tyrrhenian International Workshop on Digital Communication (IWDC): proceedings*, Sept. 2002.

[10] R. Koch, M. Pollefeys, and L. V. Gool. Multi viewpoint stereo from uncalibrated video sequences. In *Proc. ECCV'98*, number 1406 in LNCS, Freiburg, 1998. Springer-Verlag.

[11] M. Pollefeys, R. Koch, and L. J. V. Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999.

[12] J. Shi and C. Tomasi. Good features to track. In *Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, June 1994. IEEE.

[13] J. Woetzel and R. Koch. Real-time multi-stereo depth estimation on GPU with approximative discontinuity handling. In *1st European Conference on Visual Media Production (CVMP 2004), London, United Kingdom*, March 2004.