Video-based 3D Reconstruction of Moving Scenes Using Multiple Stationary Cameras[1)]

Michael Bleyer and Margrit Gelautz
Interactive Media Systems Group
Institute for Software Technology and Interactive Systems
Vienna University of Technology
Favoritenstrasse 9-11/188/2, A-1040 Vienna, Austria
e-mail: bleyer@ims.tuwien.ac.at, gelautz@ims.tuwien.ac.at

Abstract:
In this paper, we describe a system for video-based 3D reconstruction of dynamic scenes using stereo techniques, with an eye to potential applications in human motion capture. We incorporate into our approach recent research results on stereo matching which make the system efficient and produce good-quality results. The implementation is built on top of Intel's Open Source Computer Vision Library (OpenCV). Examples of 3D reconstruction results obtained from three synchronized video cameras are shown and discussed.

1 Introduction

The improvement of existing algorithms and systems for the 3D video analysis of moving scenes is important in a variety of applications including human-computer interfaces, video editing and animation, biomechanics and sports analysis, or surveillance systems. Whereas most existing studies on the stereo-based analysis of humans from video rely on conventional correlation-based techniques, we incorporate in our approach recent research results on stereo matching as described by [5]. In that study, the performance of recently developed matching algorithms is evaluated in application to synthetic data and static scenes. However, no results are reported from dynamic scenes containing humans. The authors of [4] show disparity maps of the human body that were derived by traditional correlation-based matching techniques. The quality of those stereo results is not discussed in more detail.

In our study, we describe the development of a video-based system that will be used afterwards to investigate the performance of current state-of-the-art matching algorithms in their application to dynamic scenes containing humans. The final goal is to improve those matching

[1)] This work was supported by the Austrian Science Fund (FWF) under project P15663.

algorithms for the particular purpose of reconstructing human shape and motion from video. Our system builds on components of the Open Source Computer Vision (OpenCV) Library [1], which is freely available to both scientific and commercial users.

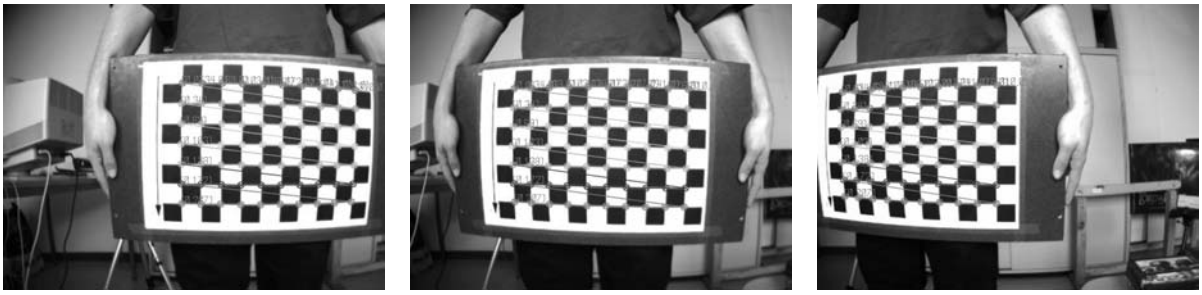
2 Method Description

In the following subsections we describe the main components of the system. We present the method used to calibrate the camera system (2.1). Camera calibration is a necessary step in order to extract metric information from 2D images. The most challenging problem in 3D reconstruction is the stereo matching process. We utilize an algorithm that operates on independent scan lines (2.2). The stereo matching algorithm uses dynamic programming to find a global optimum of a cost fuction that attempts to minimize the dissimilarity of the pixel intensities and the number and length of the occlusions. Combining the results of calibration and stereo matching allows us to calculate the 3D coordinates of corresponding 2D points (2.3). In this paper, we show results using three cameras. In principle, the implemented algorithms can be utilized to handle an arbitrary number of cameras provided that suitable mechanisms for camera synchronisation are available. The system supports input coming from video files, image files, as well as live streams.

2.1 Camera Calibration

We use the algorithm described in [7] for calculating intrinsic and extrinsic camera parameters. To calibrate the camera, the calibration routine is supplied with several views of a planar model object, or pattern, of known geometry. In OpenCV a chessboard pattern is used. Since we want the extrinsic parameters of the cameras to refer to the same world coordinate system, the calibration pattern has to be visible in every camera at the same time. An example is shown in figure 1. For every view and camera, the points on the model plane and their projections onto the image plane are passed to the calibration routine. The calibration routine calculates for each camera the intrinsic parameters, the extrinsic parameters for each image of the calibration pattern, and the coefficients of radial and tangential lens distortion.

For stereo analysis, the cameras are grouped into pairs of two cameras, which form a binocular stereo pair. In our imaging configuration, each camera, except the rightmost, and its closest neighbour to the right build such an image pair. In the example we present in section 4, we use three cameras. Therefore two image pairs are generated - one consisting of the leftmost and the camera in the middle and one consisting of the camera in the middle and the rightmost camera. Information about the camera position is derived from calibration. Since we are using a stereo matching algorithm that operates on scan lines, we need to insure a simple epipolar geometry. We achieve this by applying a method similar to the one proposed in [3].



(a) (b) (c)

Figure 1: The chessboard pattern used for calibrating the camera system. The calibration pattern has to be visible in each camera. The corners of the squares are extracted automatically. These points are passed to the calibration routine.

2.2 Stereo Analysis

We create a disparity map using the algorithm presented in [2]. It matches each pair of scan lines independently, using dynamic programming. The cost function attempts to minimize the dissimilarity of the pixel intensities and the number and length of the occlusions. We decided to use this algorithm for several reasons. First, we think that the algorithm provides a good tradeoff between the quality of the disparity maps and the computational demand, which is particularly important when dealing with videos rather than still images. Since the algorithm explicitly exploits the ordering constraint, it can handle untextured regions up to some extent, although the disparity information for large completely untextured regions is unreliable. Applying the ordering constraint also allows explicit detection of occlusion.

A drawback of this approach is that the ordering constraint does not hold true in scenes containing narrow foreground objects. Another problem of the stereo matching algorithm is inter-scan line consistency. The algorithm employs a postprocessing function that enforces consistency between scan lines. While on the one hand this postprocessing function cleans up errors, it tends to flatten the objects on the other hand. Therefore details inside the object are lost. Another crucial point is the setting of the occlusion penalty and match reward. High parameter values help in finding the correct disparity values for planar untextured regions, but details inside of an object are lost and small objects will not be found at all. Contrarily, low parameter values tend to preserve object details, but at the expense of reduced quality in regions without texture.

To speed up the stereo matching, we exploit temporal redundancies between consecutive images of a video sequence. Therefore we detect those pixels in the reference image that have changed from one frame to the next frame by computing the absolute differences of pixel intensities in subsequent frames. Pixels that have a larger difference than a specified threshold are assumed to have changed. The idea is that we want to avoid the recalculation

of disparity values for the whole scan line, if only a few pixels have changed or nothing has changed at all; instead we want to recompute disparity values only for those areas that have changed. Since for untextured moving objects only the pixels of the object's outline differ significantly from the corresponding pixel values in the previous frame, but pixels inside the object remain basically unchanged, it is not sufficient to recalculate the disparity values only for those pixels that were found to have changed. Therefore we search each scan line for the leftmost and rightmost pixel that has changed and recalculate the disparity values for every pixel between the two endpoints. Depending on the scene content, we found that this techniques can significantly speed up the computation. For example, for a scene containing one human in motion covering one third of the image, the computation is three times faster compared to the runtime behaviour of the original algorithm.

2.3 3D Reconstruction and Visualization

Combining disparity and calibration information allows us to deduce the z coordinate of a 2D point by applying elementary geometry. The reader may be referred to [6] for a detailed description. To present the points in the same coordinate system, the obtained 3D points are transformed into the 3D coordinate system of the leftmost camera, which serves as reference. We developed an efficient interactive visualization tool for viewing the generated point cloud, which allows a convenient measurement and comparison of the original and reconstructed 3D coordinates for accuracy analysis. We will extend this viewer to work on image sequences to visualize human motion in 3D space.

3 Test Data

We obtained test data by generating a set of videos using three Dragonfly IEEE-1394 colour cameras as provided by Point Grey Research. The cameras are synchronised automatically when they are on the same IEEE-1394 bus. The image resolution was set to 640×480 pixels and colour information is given in 24-bit RGB. We recorded a series of five videos showing people in motion and generated an additional synthetic video of a dynamic scene.

4 Tests and Results

Figures 2 - 6 show the steps performed to reconstruct the 3D scene. Figure 2 shows the input images that were taken simultaneously by 3 Dragonfly cameras. The 3 images form 2 stereo pairs, which we call stereo pair A, consisting of images (a) and (b), and stereo pair B with images (b) and (c). The cameras exhibit significant lens distortion. In the first step, the images are corrected for lens distortion. Each stereo pair is transformed into epipolar geometry. This means every point in one image lies on the same vertical scan line in the other

(a) left camera (b) middle camera (c) right camera

Figure 2: The input images taken by 3 synchronized video cameras.

image. Figure 3 shows stereo pair A after being corrected for lens distortion and transformed into epipolar geometry.

Figure 4 shows the disparity maps computed from stereo pairs A and B. In figure 5, the right images of each stereo pair are reconstructed by displacing the pixels of the left images using the information of the disparity map in figure 4. A comparison of figure 3 (b) and figure 5 (a) gives information on the quality of the disparity map shown in figure 4 (a). Black pixels along the right object borders correspond to pixels that are occluded in the left image.

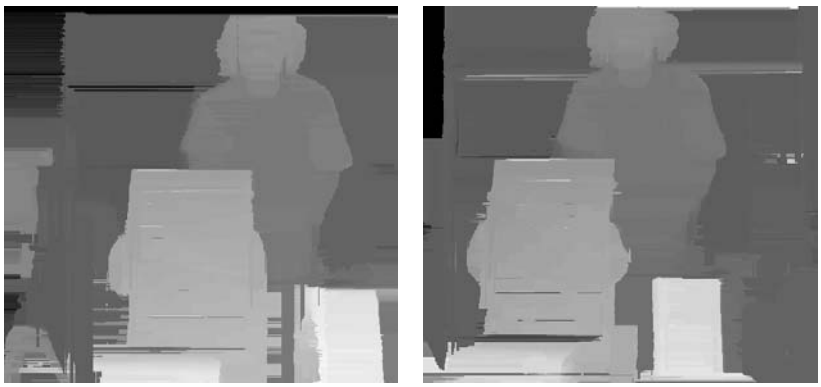
For each image pair, the 3D coordinates are calculated and transformed into the coordinate system of the leftmost camera. The generated point cloud is shown in figure 6. In figure 6 (a), every 3D point is assigned the colour of the corresponding pixel in its reference (i.e. left) image. The 3D points obtained from stereo pair B are superimposed onto the points from stereo pair A. Figure 6 (b) shows the contribution of each camera pair to the point cloud: points derived from camera pair A are coloured red (dark) - points derived from camera pair B are coloured green (bright). The superimposed coordinate values are given in meters. The reconstructed metric information corresponds to distances in the real world. Missing information in the 3D reconstruction is mainly due to occlusions in the original images.

5 Summary and Outlook

We presented a video-based system for 3D reconstruction of moving scenes, which we tested using a set of 3 synchronised cameras. The incorporation of recent results on stereo matching creates a system that produces good-quality results and keeps the computational effort relatively low. Further work will concentrate on improving the stereo matching algorithm. The system will form the basis for future work on model-based 3D reconstruction of human body shape and motion.

(a) (b)

Figure 3: Stereo image pair A from figures 2 (a) and 2 (b) corrected for lens distortion and transformed into epipolar geometry.

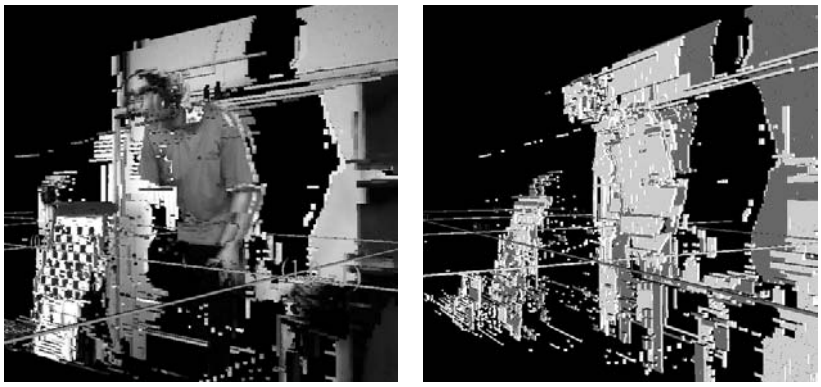


(a) (b)

Figure 4: The computed disparity maps of stereo pairs A (left) and B (right). Brighter objects are closer to the camera.

(a) (b)

Figure 5: The right image of each stereo pair reconstructed by displacing the pixels of the left images using the information of the corresponding disparity map. Subfigures (a) and (b) were derived from stereo pairs A and B, respectively.



(a) (b)

Figure 6: The reconstructed 3D point cloud. Subfigure (a) shows the intensity values taken from the original images. In subfigure (b), the intensity values indicate the contributions from the 2 camera pairs.

References

[1] Intel open source computer vision library, beta 3.1. http://www.intel.com/research/mrl/research/opencv/.

[2] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo. *International Journal of Computer Vision*, 35(3):269–293, December 1999.

[3] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint.* pp. 188–189, MIT Press, 1993.

[4] R. Plänkers and P. Fua. Articulated soft objects for multi-view shape and motion capture. *To appear in IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2003.

[5] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/2/3):7–42, April-June 2002.

[6] M. Sonka, V. Hlavac, and R. Boyle. *Image processing, analysis, and machine vision.* p. 460, Chapman and Hall Computing, 1999.

[7] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.