Temporally Consistent Disparity Maps from Uncalibrated Stereo Videos

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

Abstract

We address the problem of computing a sequence of dense disparity maps from two synchronized video streams recorded by slightly displaced cameras. Generating such disparity videos is becoming increasingly important in the light of new autostereoscopic displays and novel viewpoint applications. We propose a good-quality, computationally fast and easy-to-use solution to accomplish this task.

This paper describes the four major steps of our 2D to 3D video conversion procedure. (1) The user segments the video into its scenes. (2) For each scene, we rectify the uncalibrated stereo pairs so that correspondences lie on the same horizontal scanline. (3) A fast and accurate dynamic programming-based stereo matcher is then applied to compute a dense disparity map for each stereo pair. (4) We perform temporal smoothing on the computed disparity sequence to reduce the disparity flickering problem. All of these functionalities can be accessed via an easy-to-use Graphical User Interface, which makes our conversion procedure applicable even for technical unskilled users. We demonstrate the good quality of our results using various challenging real-world stereo streams.

1. Introduction

Autostereoscopic displays have recently made a big step in the quality of 3D impression provided to the user. Together with constantly decreasing prices, this is starting to make them attractive also for the mass market. In addition to a color image, displays such as the Philips WOWvx screen [1] require a corresponding depth map of the scene for enabling 3D viewing. These depth maps are used to synthesize arbitrary stereo views as they would appear from different viewing angles (i.e. novel viewpoint generation). The user can therefore walk in front of the display and get a perspectively correct 3D impression of the scene from its current viewing point.

One major problem that currently hinders the spread of such displays is the difficulty of content creation. While for artificial content (e.g. animation movies) depth maps can easily be made available as byproduct of the production process, this is clearly more challenging for videos of real scenes. Apart from a rather expensive commercial solution (Philips Blue Box), there does currently not exist a feasible method for a standard user to perform the 2D to 3D conversion in an automatic manner.

In the context of prior work, there exists a vast amount of literature on the stereo matching problem for single image pairs. The reader is referred to [10] for a review and evaluation of existing stereo techniques. However, the problem of matching stereo videos - which also includes a temporal component - is studied in much less detail. In this context, the work of Zitnick et al. [12] computes disparity maps in multi-view video sequences for generating a novel viewpoint system. However, the authors do not consider the temporal relationship between disparity frames. Larsen et al. [9] use optical flow to propagate disparities among consecutive frames of a stereo video sequence. However, the belief propagation-based optimization makes their algorithm relatively slow. Recently, Sizintsev and Wildes [11] have proposed to model temporal continuity without explicitly recovering the optical flow via using stequels as matching primitives. In contrast to prior work, our goal is to provide a good-quality and computational efficient method for performing the 2D to 3D video conversion. Our conversion procedure is intended to be used also by people without computer vision background.

2. Method

Figure 1 illustrates the workflow of our approach. Our method accepts a stereo video as input. In the first step, this video is divided into its individual scenes. The stereo pairs of each scene are then rectified so that pixel correspondences lie on the same horizontal scanline in both images. This rectification is required by our stereo matcher. Given the rectified images, the stereo matcher produces a dense disparity map that represents the distance of each pixel to the camera. Finally, we apply smoothing on the disparity map sequence to enforce temporal consistency. This smoothing procedure aims at overcoming the disparity flickering problem. We describe each of these steps in more detail in the remainder of this section.

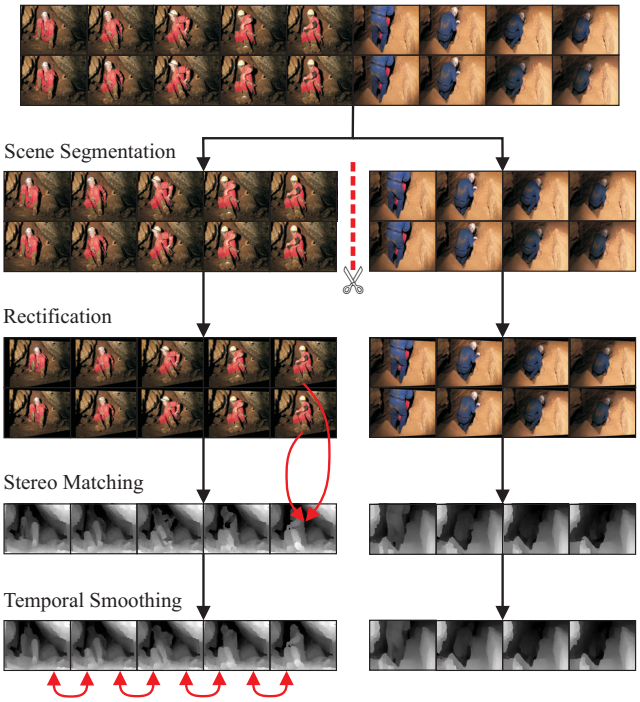

Figure 1. Overview of our method. More information is found in the text

Figure 2. Scene segmentation. The user is asked to set the shot boundaries by clicking the scissors button.

2.1. Scene Segmentation

The input to our conversion is formed by two video files representing the left and right cameras, respectively. Once the user has loaded the video, he/she is asked to specify the shot boundaries as is shown in figure 2. We are planning to automate this process in future work by using one of many existing shot boundary detection methods (e.g. [5]).

The reasons for dividing the movie into its scenes are threefold. First, it makes sense to run an individual calibration procedure for each scene as is described in section 2.2. Second, we use different disparity ranges for each scene in the stereo matching step of section 2.3. Third, we aim to avoid smoothing over shot boundaries in the temporal smoothing step of section 2.4.

2.2. Rectification

In the ideal case, one would fully calibrate the stereo camera system before recording the 2D videos. The calibration parameters would then be provided as input to our conversion method. However, it is difficult to perform stereo calibration for a standard user without experience in computer vision. Moreover, there is also a large number of already existing stereo streams for which the calibration information is simply not available. We have therefore decided to incorporate uncalibrated rectification into our method so that the rectification matrices are computed solely from the image content of the stereo video.

Our rectification procedure assumes that internal and external camera parameters remain constant within a scene. In particular, this means that the relative positioning of the cameras to each other (baseline, camera angles) should not vary within the scene. Assuming constant calibration parameters, we can rectify all stereo pairs of the scene using the same rectification matrices.

To compute these rectification matrices, we determine a set of correspondences between left and right images using the computational efficient SURF features [2]. In principle, it is sufficient to use the correspondences of only a single image pair as input to the subsequent rectification algorithm. However, for the sake of robustness, the user can specify multiple stereo pairs (e.g. 10) from which the correspondences are gathered. In our experiments, the capability to use more than one image pairs in our method has considerably improved our rectification results.

The correspondences form the input for a rectification algorithm. We have first experimented with the standard method of Hartley and Zisserman [7]. However, we have finally decided to implement the rectification algorithm of Fusiello and Irsara [6], as we have found that it produces accurate rectified views of smaller distortions.

2.3. Stereo Matching

We apply a dense stereo matching algorithm that is based on the method of [3]. This algorithm uses dynamic programming for disparity optimization. However, as opposed to conventional dynamic programming-based methods (e.g. [4]), it does not operate on scanlines, but on trees that include horizontal and vertical smoothness edges. The algorithm can therefore overcome the scanline streaking problem, which is the inherent disadvantage of classical dynamic programming methods.

From a practical point of view, the applied stereo method is attractive, since it is capable of producing disparity results of similar quality in comparison to state-of-the-art techniques such as graph-cuts or belief propagation. In contrast

to these state-of-the-art techniques, our stereo algorithm offers the advantage of greatly reduced processing time. Even for relatively high resolutions, results for a stereo pair are computed in approximately a second. We are currently working on a graphics card implementation to achieve real-time frame rates.

We have extended the method of [3] to improve its applicability on real-world stereo pairs. Our first extension allows coping with radiometric distorted stereo images. For example, we can handle the case in which the left image is darker than the right one due to different illumination conditions under which the left and right images have been recorded. Our second extension is the inclusion of sub-pixel support. Sub-pixel accuracy is important for reconstructing detailed disparity surfaces, especially if there is only small parallax between left and right images.

One critical point in stereo matching is the setting of algorithm parameters. The first parameter that we have to determine is the disparity range, i.e. the range in which the algorithm searches for correspondences. This parameter is found via user input (figure 3a). First, the user has to overlay the images of a stereo pair so that the background is "in focus". This defines the minimum allowed disparity. This user input is then also required for the image foreground (maximum allowed disparity). Moreover, the user can "tune" the stereo matching results. We therefore generate nine different disparity results by changing the smoothness parameter of our algorithm. The user then selects the best disparity result (figure 3b) and the smoothness parameter is adjusted accordingly. However, we have noticed that our stereo algorithm is relatively insensitive to different parameter settings so that the default parameters are typically sufficient for obtaining good-quality results.

2.4 Temporal Smoothing

In principle, the disparity maps generated in the previous section can already be used as the result of the 3D conversion procedure. However, the resulting disparity sequence is, in general, not temporally smooth. A common artifact is the so-called disparity flickering, which is caused by erroneously computed abrupt disparity changes over time. Especially in the context of autostereoscopic displays, artifacts of this type are extremely disturbing for the viewer. We tackle the disparity flickering problem using the temporal smoothing algorithm described in the following.

The median filter represents a natural choice for the desired temporal smoothing. In a naive implementation, one can smooth the disparity $d_{p,f}$ of pixel p in the fth frame by computing the array of disparities D:

$$D = \{d_{p,i} | f - \sigma \leq i \leq f + \sigma\}. \quad (1)$$

Here, σ represents a parameter that defines the smoothing strength. The smoothed disparity $d'_{p,f}$ is then derived by

$$d'_{p,f} = \mathrm{med}(D) \quad (2)$$

where $\mathrm{med}()$ denotes a function that computes the median.

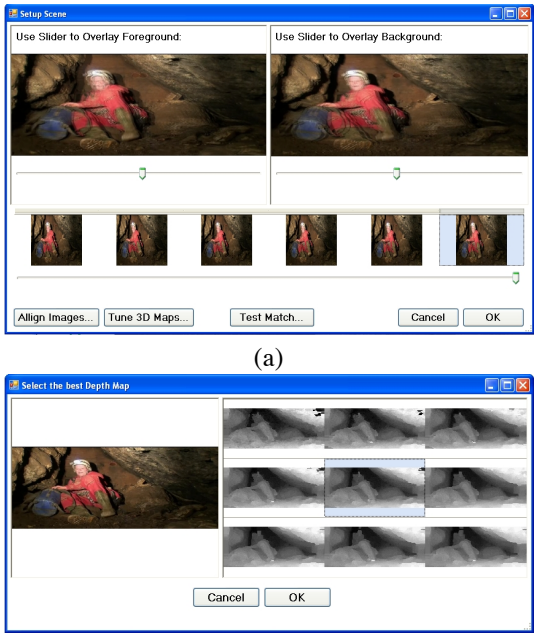

Figure 3. Adjusting the parameters of the stereo matcher. (a) The user defines the disparity range by overlaying left and right images. (b) The user sets the algorithm's internal parameters by selecting one of the nine disparity maps.

However, the problem of this method is that it fails if there is motion in the disparity sequence (e.g. consider a camera pan). To overcome this problem, we compute the optical flow in the sequence of left input images.[1] More precisely, we compute the optical flow between frames 1 and 2, frames 2 and 3 and so on. To avoid that the optical flow computation becomes the bottleneck in our conversion procedure, we apply a fast implementation of the method of Horn and Schunck [8]. Instead of using stationary pixels, we can now exploit the optical flow results to build the array D of equation (1) in a more correct manner. To compute D, we trace the pixel over time using the optical flow vectors. Hence, we can cope with image motion in our smoothing procedure. Analogously to above, the smoothed disparity is then derived by computing the median in the array D (equation (2)). The user can change the amount of temporal smoothing by changing the parameter σ of equation (1). By default, the parameter σ is set to 3.

3. Results

In order to make our method applicable for users with relatively little technical skills and to simplify the handling of the huge amount of image data, we have built an easy-to-use Graphical User Interface. The main window of this

[1] The left images are chosen, because also the disparity maps are computed in the geometry of the left image.

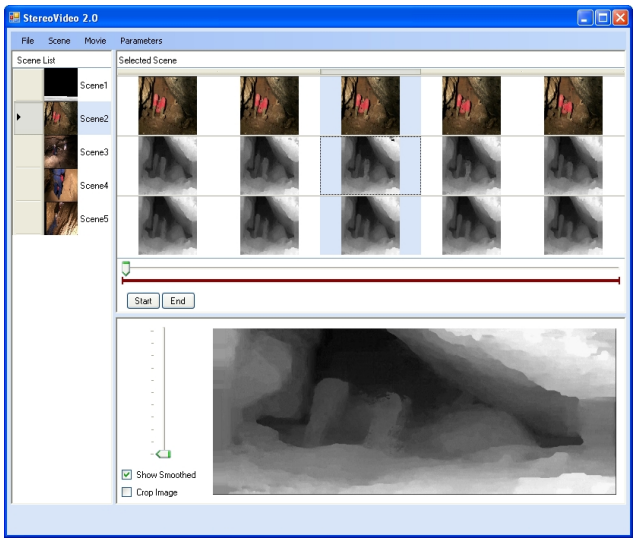

Figure 4. The main window of the Graphical User Interface.

interface is shown in figure 4. From this form the user can access all of the described functionality (scene segmentation, rectification, stereo matching and temporal smoothing) and inspect the generated disparity maps.

We have tested our 2D to 3D video conversion method using rather large sequences of 5 minutes video content at 25 frames per second. The image resolution of these sequences ranges from 720 $\times$ 288 to 720 $\times$ 576 pixels. At these resolutions, computation of a disparity map takes approximately a second.

Figure 5 shows the results of our 2D to 3D conversion on some selected real-world sequences. More precisely, the figure shows the left images of the stereo sequences along with the computed disparity maps after temporal smoothing. It can be observed that the results of our disparity computation are of good quality and temporally consistent.

We have also tested viewing the disparity videos on the Philips WOWvx display [1]. The depth layers appeared to be correct and the object boundaries seemed to be accurately captured. Note that especially the accuracy of depth boundaries is important for satisfactory viewing experience on the autostereoscopic display.[2] Disparity flickering has almost not been visible in the computed disparity sequences.

4. Conclusions and Future Work

We have presented a method for generating temporal-consistent disparity sequences from uncalibrated stereo video streams. The advantages of the proposed method include the good quality of results, fast computation times and an easy-to-use interface for standard users without computer vision background. We have described the main steps of our 2D to 3D conversion procedure. This includes interactive scene segmentation and our method for performing uncalibrated rectification. We have then provided insights into our stereo matching procedure and presented our technique for enforcing temporal consistency in the disparity sequence. We have demonstrated the good results of our conversion procedure using various real-world stereo sequences.

Future work will concentrate on further improving the computational performance of our stereo matching algorithm to achieve real-time frame rates. We are also planning to test alternative optical flow methods for the temporal smoothing step, since the results of our current optical flow implementation are of moderate quality.

[2] In contrast to this, disparity errors in untextured regions only play a minor role.

Acknowledgments

Michael Bleyer is financed by the Austrian Science Fund (FWF) under project P19797.

References

[1] Philips wowvx. http://www.philips.com/3Dsolutions.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features", *Computer Vision and Image Understanding*, 110(3), 2008, pp. 346–359.

[3] M. Bleyer and M. Gelautz, "Simple but effective tree structures for dynamic programming-based stereo matching", In *VISAPP*, volume 2, pp. 415–422, 2008.

[4] A. Bobick and S. Intille, "Large occlusion stereo", *International Journal of Computer Vision*, 33(3), 1999, pp. 181–200.

[5] M. Cooper, T. Liu, and E. Rieffel, "Video segmentation via temporal pattern classification", *Transactions on Multimedia*, 9(3), 2007, pp. 610–618.

[6] A. Fusiello and L. Irsara, "Quasi-euclidean uncalibrated epipolar rectification", In *International Conference on Pattern Recognition*, pp. 1–4, 2008.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN: 0521623049, 2000.

[8] B. Horn and B. Schunck, "Determining optical flow", *Artificial Intelligence*, 17, 1981, pp. 185–203.

[9] E. Larsen, P. Mordohai, M. Pollefeys, and H. Fuchs, "Temporally consistent reconstruction from multiple video streams using enhanced belief propagation", In *International Conference on Computer Vision*, 2007.

[10] D. Scharstein, "View synthesis using stereo vision", *Lecture Notes in Computer Science (LNCS)*, 1583, 1999.

[11] M. Sizintsev and R. Wildes, "Spatiotemporal stereo via spatiotemporal quadric element (stequel) matching", In *Conference on Computer Vision and Pattern Recognition (to appear)*, 2009.

[12] L. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation", *ACM Transaction on Graphics*, 23(3), 2004, pp. 600–608.

Figure 5. Results of our method. For each sequence, we show the left image of the stereo pair along with the computed temporally smoothed disparity map.