Synthesis of Stereoscopic 3D Videos by Limited Resources of Range Images

Xiaoyi Jiang\(^1\) and Martin Lambers\(^2\)

\(^1\)Department of Mathematics and Computer Science, University of Münster, Germany
Email: xjiang@math.uni-muenster.de

\(^2\)Computer Graphics Group, University of Siegen, Germany
Email: lambers@fb12.uni-siegen.de

Abstract

Recently, depth-image-based rendering has been suggested for 3D video creation, in which a monoscopic video stream is recorded along with a depth (range) video stream and a stereoscopic video stream is then generated at the user end. The fundamental assumption of this framework is the full availability of range images at video rate. In this work we alleviate this hard demand and assume that only limited resources of range images are available, i.e. corresponding range images exist for some, but not all, color images of the monoscopic video stream. We propose to synthesize the missing range images between two consecutive range images. Our approach eases the recording of 3D material by using less expensive range sensors and enables to enhance existing 2D video material with 3D effect by limited manual overhead. Experiments on real videos have demonstrated very encouraging results. Especially, one 3D video was generated from a 2D video without any sensory 3D data available at all. In a quality evaluation using an autostereoscopic 3D display the test viewers have attested similar 3D video quality for our synthesis technique and rendering based on depth ground truth.

1. Introduction

Early approaches to 3D video are based on an end-to-end stereoscopic video chain, i.e. capturing, transmitting and display of two separate video streams, one for each eye. Recently, an advanced concept of depth-image-based rendering (DIBR) has been proposed [3, 9]. 3D video systems based on DIBR require a single stream of monoscopic images and a second stream of depth (range) images which convey per-pixel depth information. Given the two image streams, a high-quality stereoscopic stream for any nearby viewpoint can be synthesized.

Compared to the end-to-end stereoscopic video stream, this concept has a number of advantages (see [3] for full details): backward compatibility with existing 2D video systems; flexibility (optimal 3D effects customized to different 3D displays and user needs; support of multiview 3D displays); efficiency (coding and transmission of the range video stream cheaper than a monoscopic video stream). The most important components of DIBR-based 3D videos are: content generation, coding, transmission, virtual view synthesis and 3D display. Our current work is devoted to content generation.

1.1. 3D content generation

The fundamental assumption in DIBR-based 3D videos is the availability of range images at video rate. This can be achieved by a real-time 3D camera [3]. The practical value of this approach, however, is still quite limited. Currently, only very few range cameras deliver real-time range videos and their use is typically restricted by limiting factors such as operational environment (indoor, outdoor) and ranging area (angular field of view, depth of field). In addition high-rate range cameras tend to be expensive and this hinders their use in a broad range of applications.

Alternatively, depth information can be computed from a single image [1]. Several classes of shape-from-X methods follow this goal, for instance shape-from-shading [8]. Their practical value, however, is still extremely limited.

A third way of 3D content generation is recovery from 2D videos. Despite of the advances in the past, automatic 3D reconstruction remains a tough challenge [6, 7]. Some approaches require an object segmentation [5] which causes additional uncertainty to the difficult recovery task.

1.2. Our approach

We assume that only limited resources of range images are available, i.e. corresponding range images exist for some, but not all, color images of the monoscopic video stream, and propose to synthesize the missing range images between two consecutive range images. This allows to: a) Ease the recording of 3D material. Instead of using expensive video-rate range sensors it is possible to use cheaper sensors that generate less range images and complete the missing range images automatically; b) Enhance existing 2D video material with 3D effects by automatically completing depth information from a few, possibly manually
created, range images. Given the vast amount of existing 2D material, this is an important application.

The basic idea of our approach is to estimate motion in the monoscopic video stream and to apply this motion information for synthesizing the missing range images. The technical details are described in next section. Given a color image and its corresponding range image, a stereo pair can be synthesized by a special 3D image warping technique [3, 10]. Experimental results are reported in Section 3. Finally, we conclude the paper with some discussion.

2. Synthesizing range images

It is assumed that a monoscopic (color) video stream is given by \( n \) frames \( F_0, \ldots, F_{n-1} \), along with depth information \( D_0 \) and \( D_{n-1} \) for the first and the last frame. The goal is to expand the depth information from \( D_0 \) and \( D_{n-1} \) so that a complete set of range images \( D_0, \ldots, D_{n-1} \) is available for the subsequent depth-image-based rendering step.

2.1. Range Image Synthesis by Depth Tracking

The basic idea is to track each point in the scene as it moves to different pixel positions from frame to frame. For a point \( P \) at position \( p_i \in F_i \), \( 0 < i < n - 1 \), the corresponding positions \( p_0 \in F_0 \) and \( p_{n-1} \in F_{n-1} \) are then known, and therefore also the associated depths \( d_0 \) and \( d_{n-1} \). The depth \( d_i \) required for the unknown depth map \( D_i \) can then be computed by an interpolation of \( d_0 \) and \( d_{n-1} \).

To be able to compute the position \( p_{i+1} \in F_{i+1} \) of a point \( P \) with a given position \( p_i \in F_i \), it is necessary to know the forward motion vector \( f_i(p_i) \): \( p_{i+1} = p_i + f_i(p_i) \). Similarly, the position \( p_{i-1} \) in \( F_{i-1} \) of the point can be computed when the backward motion vector \( b_i(p_i) \) is known: \( p_{i-1} = p_i + b_i(p_i) \). To handle all points of the scene, the forward and backward motion vector fields \( f_0, \ldots, f_{n-2} \) and \( b_1, \ldots, b_{n-1} \) that contain motion vectors for each pixel position must be computed. Note that it is not enough to know the motion vectors of only one direction, because the per-pixel motion between two frames is not bijective. The complete process of depth tracking from motion analysis is illustrated in Figure 1. Details will be given in Section 2.2.

Based on the depth tracking the missing range images are synthesized in the following way. A point \( P \) with a given position \( p_i \) in frame \( F_i \) is tracked backwards to some position \( p_0 \) in frame \( F_0 \) and forwards to some position \( p_{n-1} \) in frame \( F_{n-1} \). We distinguish between three cases: 1) Both backward and forward tracking are successful: A linear interpolation of the corresponding depth values \( d_0 \) and \( d_{n-1} \) is performed to determine \( d_i \); 2) It can only be tracked to one of both end frames: \( d_i \) is set to the depth value at this end frame, thus assuming that it remains constant over time; 3) It can neither be tracked to \( F_0 \) nor to \( F_{n-1} \): the depth \( d_i \) is arbitrarily set to "far". To avoid too frequent occurrences of such untrackable situations, it is necessary that the observed scene does not change too much, so that roughly the same objects are visible in \( F_0 \) and \( F_{n-1} \), albeit at different positions. If this requirement is not fulfilled, the scene may have to be divided into sub-scenes.

2.2. Details of Depth Tracking

The key part of depth tracking is the per-pixel motion estimation: The more precise it is, the better the depth approximation. We have experimented with two approaches to compute motion vector fields: optical flow and block matching.

Optical flow: The per-pixel motion vector fields are in fact dense optical flow fields. Optical flow computation techniques can be classified into local methods that optimize some local energy-like expression and global strategies that attempt to minimize a global energy functional. Examples are the Lucas-Kanade method and the classical work of Horn-Schunck, respectively.

Local and global methods have quite different smoothing effects. Consequently, it is desirable to combine them in order to design novel approaches of both the high robustness of local methods and the full density of global techniques. A recent combination scheme is proposed in [2]. This technique and the methods of Lucas-Kanade and Horn-Schunck are used in our experiments.

Block matching: Block matching techniques are commonly used in feature tracking applications and in stereo correspondence search: to find a match for the pixel at position \( p \) in frame \( F_0 \) at some position \( q \) in frame \( F_1 \), a block of size \((2k+1) \times (2k+1)\) around \( p \) is examined, and the best match for this neighborhood is searched in \( F_1 \). If \( \hat{q} \) is the position of the match candidate currently under consideration, then its matching costs are defined as:

\[
C(p, \hat{q}) = \sum_{r=-k}^{k} \sum_{c=-k}^{k} c(p+(r,c), \hat{q}+(r,c))
\]  

The candidate position \( \hat{q} \) with the lowest matching costs wins. The cost function \( c \) differs between various block matching variants.

![Figure 1. Depth tracking from motion analysis](image_url)
One popular cost function uses the absolute difference of pixel values:

\[ c_{\text{SAD}}(p, q) = |F_0(p) - F_1(q)| \]

This expression can be easily extended to handle color images. The YUV color space is widely used in video processing applications. The Y component represents the brightness of a point, and the U and V components define its hue and saturation. Thus, the following term can be used:

\[ c_{\text{SAD}}(p, q) = L \cdot |Y(F_0(p)) - Y(F_1(q))| + (1 - L) \cdot \frac{1}{3} \left( |U(F_0(p)) - U(F_1(q))| + |V(F_0(p)) - V(F_1(q))| \right) \]

Each of the components Y, U, V is expected to be in \([0, 1]\) in this equation. L is the luminance weight: It determines how much influence luminance differences should have in comparison to color differences.

To increase the computational efficiency and simultaneously reduce the uncertainty of the method, the SAD block matching variant for depth tracking uses the following matching cost function, which is an extension of Eq. (1):

\[
C_{\text{SAD}}(p, \hat{q}) = D \cdot \frac{\text{distance}(p, \hat{q})}{k} + (1 - D) \cdot \sum_{r=-k}^{k} \sum_{c=-k}^{k} c_{\text{SAD}}(p + (r, c), \hat{q} + (r, c))
\]

(2)

The additional term is a distance penalty: Larger motion vectors cause higher matching costs. The parameter \(D\) determines the balance between the distance penalty and the sum of absolute pixel value differences. It should be small (\(\leq 0.1\)), because large distance penalties corrupt the results.

The introduction of a distance penalty for the candidate position \(\hat{q}\) has a nice side-effect of allowing to shortcut the computation of \(C_{\text{SAD}}\): If motion vectors \(\hat{q} - p\) are tested in the order of ascending length (see Figure 2), then the full costs \(C_{\text{SAD}}\) do not have to be computed if the distance penalty alone is already greater than the total cost of the previously tested optimal vector. This is frequently the case in background areas of scenes: since no object moves, a very good match with \(C_{\text{SAD}}\) near zero is already found at a motion vector near \((0, 0)\). Motion vectors tested at a later time then exceed these very low costs already because of their length, so the sum of differences does not need to be computed. The distance penalty also reduces the uncertainty, for example in areas with periodic textures, where good matches are found at multiple positions.

Consistency check: A postprocessing step often used for stereo correspondence search is adapted to improve the motion vector fields for depth tracking, regardless of the motion estimation method they were created with.

To catch unreliable motion vectors, the motion estimation is done in both directions: from \(F_0\) to \(F_1\), leading to the vector field \(f\), and from \(F_1\) to \(F_0\), leading to the vector field \(b\). The reliability of a motion vector \(v\) in \(f\) will be high if the corresponding motion vector in \(b\) points back to the position of \(v\) or near to it. A threshold \(t\) determines the maximum allowed difference for vectors to be considered reliable. If the difference is greater, the vector \(v\) in \(f\) is marked as unreliable. In a second step, all unreliable vectors in \(f\) are replaced by interpolating neighboring reliable vectors. This is done in a way that ensures that vectors with a high number of reliable neighbors are replaced first, to avoid propagating errors as much as possible. The result is an improved vector field \(f^*\). By swapping the roles of \(f\) and \(b\), the same can be done with \(b\), leading to the improved field \(b^*\). Since the motion estimation for depth tracking has to be done in both directions anyway, the consistency check imposes little additional computation costs.

3. Experimental results

3.1. Test material

Three example video scenes were used:

- **Interview**: This scene shows a man and a woman sitting at a table and having a chat. It is roughly ten seconds long (251 frames) and has a resolution of 720 × 576 pixels, which corresponds to the European PAL TV standard.
- **Orbi**: This scene shows an astronaut working on a space shuttle, with the earth in the background. It is eighteen seconds long (451 frames). The resolution is 528 × 360.
- **Nasa**: This scene shows a space shuttle, with the earth in the background. It is ten seconds long (451 frames). The resolution is 528 × 360.

The scenes **Interview** and **Orbi** (see Figure 3 for two images of each video) have known depth maps for each video frame, recorded with a real-time depth sensor. The real depth data can be used as ground truth when evaluating computed depth data.

The scene **Nasa** is a part of a NASA mission video. It is a conventional 2D video without any depth data. Three sim-

---

Figure 2. Order in which motion vectors are tested: Example for 25 vectors at a maximum distance \(d = 2\)

---

Though not in the direct way as for example the HSL color space does.
plastic, qualitative range images were created manually with minimal efforts: one for the first frame, one for the middle frame, and one for the last frame, resulting in a distance of 225 frames between two known depth maps.

3.2. Evaluation of depth tracking

The three optical flow methods and the SAD block matching method were tested to find the best per-pixel motion estimation method for depth tracking. In the following we report some results of evaluating the depth tracking quality based on the Interview video; see [4] for more details.

The frames $F_0, \ldots, F_{250}$ (ten seconds) of the Interview scene were extracted, along with real depth data $D_0, D_{25}, \ldots, D_{250}$ for every 25th frame. The missing depth maps were computed using depth tracking with the motion estimation method under examination.

Figure 4 shows two frames $D_0$ and $D_{25}$ and, for illustrating the motion between these two frames, the difference image. In addition we can see the depth map $D_{12}$ produced with the four motion estimation methods. For Horn/Schunck (HS), Lucas/Kanade (LK), and the combined local/global (CLG) method, some errors in motion estimation are clearly visible. For example, around the head of the man, the effects of linear interpolation outweigh the effects of motion estimation. The SAD block matching method seems to deliver the best motion estimation, resulting in the most accurate depth map. Around the head of the man, for example, only small effects of linear interpolation are visible; for the most part, its motion was followed correctly.

The differences between ground truth and the computed depth maps can be used as an error measurement:

$$E_{DT} = \frac{\sum_{i=0}^{N-1} \sum_{y=0}^{M-1} |D_i(x,y) - GT_i(x,y)|}{nNM}$$

This average depth error value is in $[0, 255]$ (the smaller, the better). Table 1 shows the results for the complete series of depth maps from which the examples in Figure 4 were taken. The depth maps with the lowest errors according to this measurement are the ones from the SAD block matching method. This confirms the first impressions from Figure 4. Similar behavior has been consistently observed for the videos Interview and Orbi. Currently, we still cannot make conclusive claims. Other videos may well show another picture, i.e., different video material may require different motion analysis methods.

3.3. Rating of 3D video quality

Several 3D video variants of each of the videos Interview, Orbi, and Nasa were computed from different depth data, and prepared for display on a 2018XLQ 19” 3D monitor from DTI\(^3\), which is an autostereoscopic display, al-

\(^3\)http://www.dti3d.com/
following 3D viewing experiences without the need of wearing any glasses. Interested readers can find a summary of (mostly commercially available) autostereoscopic displays at http://www.stereo3d.com/displays.htm.

A group of 10 test viewers was asked to rate both the image quality and the quality of the 3D effect of each variant, on a scale from 0 ("very bad" or "nonexistent") to 10 ("excellent"). A note handed out to each test viewer clarified that image quality means the absence of noise and distortion in the image, and 3D effect quality means the impression of real depth. The viewers did not have any experience with 3D videos, and they did not know anything about the nature of the videos and their variants.

The following three variants were tested:

1. **2D**: The original 2D video. Presenting this variant allows to measure the impact that depth-image-based rendering has on image quality.

2. **Ground Truth**: A 3D video based on real depth maps. This variant is expected to show the best results in terms of 3D effect quality. For the *Nasa* scene, this option was not used due to the lack of real depth data.

3. **Depth Tracking**: A 3D video based on depth data that was computed using the depth tracking method. For *Interview* and *Orbi*, every 25th depth map from ground truth was used as initial depth data. For *Nasa*, the three artificial depth maps were used.

The rating results are shown in Figure 5. Comparing the rating of the 2D variant with the results of other variants shows that the image quality always suffers a little from depth-image-based rendering. This effect may be reducible by choosing better parameters for the rendering step, but this was not subject of the test. Most viewers noticed the absence of any 3D effect in the 2D variant. Remarkably, the 3D videos synthesized by depth tracking was rated similar to those from the ground truth depth information with respect to both image quality and 3D effect.

### 4. Conclusions

In this paper we have considered a range image synthesis technique for reducing the need of full availability of a range video stream in DIBR-based 3D video creation. Our approach eases the recording of 3D material by using less expensive range sensors and enables to enhance existing 2D video material with 3D effect by limited manual overhead. Experiments on three videos have demonstrated very encouraging results. Especially, one 3D video was generated from a 2D video without any sensory 3D data available at all. In all cases the test viewers have attested similar 3D video quality for our synthesis technique and rendering based on depth ground truth.

### References


