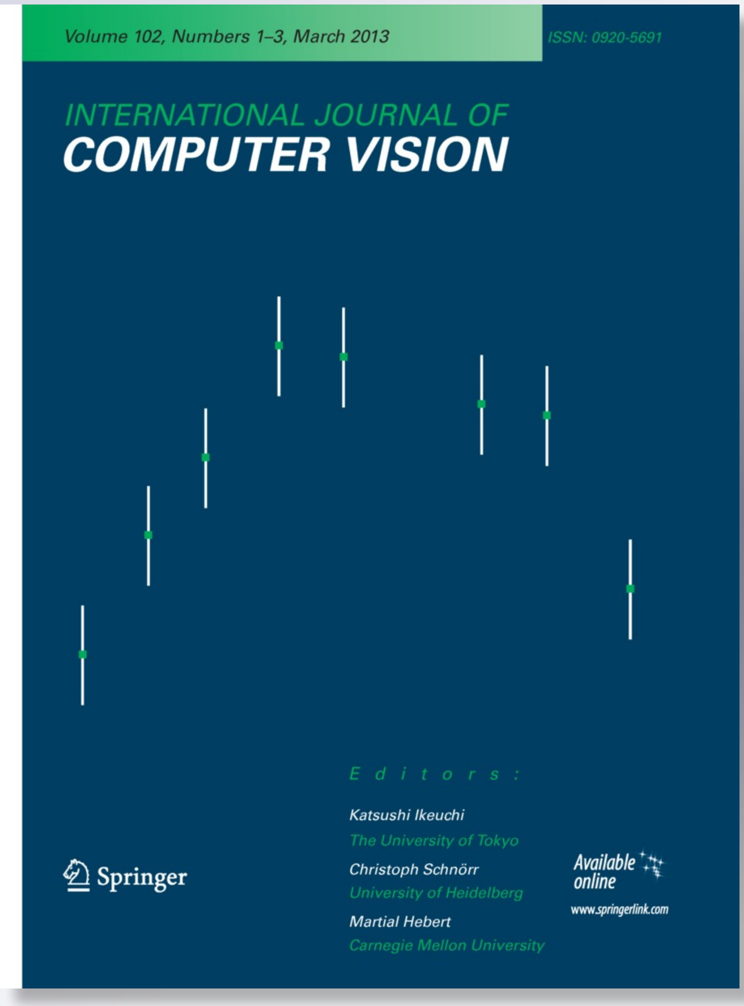
International Journal of Computer Vision

ISSN 0920-5691
Volume 102
Combined 1-3

Int J Comput Vis (2013) 102:293-307
DOI 10.1007/s11263-012-0593-9



DOI 10.1007/s11263-012-0593-9

Depth Mapping for Stereoscopic Videos

Tao Yan · Rynson W. H. Lau · Yun Xu · Liusheng Huang

Received: 5 November 2011 / Accepted: 19 October 2012 / Published online: 9 November 2012

Abstract Stereoscopic videos have become very popular in recent years. Most of these videos are developed primarily for viewing on large screens located at some distance away from the viewer. If we watch these videos on a small screen located near to us, the depth range of the videos will be seriously reduced, which can significantly degrade the 3D effects of these videos. To address this problem, we propose a linear depth mapping method to adjust the depth range of a stereoscopic video according to the viewing configuration, including pixel density and distance to the screen. Our method tries to minimize the distortion of stereoscopic image contents after depth mapping, by preserving the relationship of neighboring features and preventing line and plane bending. It also considers the depth and motion coherences. While depth coherence ensures smooth changes of the depth field across frames, motion coherence ensures smooth content changes across frames. Our experimental results show that the proposed method can improve the stereoscopic effects while maintaining the quality of the output videos.

Keywords Stereoscopic videos · Depth mapping · Image warping · Video processing

T. Yan · Y. Xu · L. Huang
University of Science and Technology of China, Hefei, China
e-mail: yantao@mail.ustc.edu.cn

Y. Xu
e-mail: xuyun@ustc.edu.cn

L. Huang
e-mail: lshuang@ustc.edu.cn

T. Yan · R. W. H. Lau (✉)
City University of Hong Kong, Kowloon Tong, Hong Kong
e-mail: rynson.lau@cityu.edu.hk

1 Introduction

Stereoscopic 3D images and movies have become very popular in recent years. More and more movies are being produced in 3D, which also drives the popularity of 3D TVs and displays. Some mobile phone companies have even begun to produce cell phones with 3D displays.

In general, each stereo image contains two regular 2D images captured from the same scene at the same time but from slightly different viewing locations. When a stereo image/video is displayed on the screen, with appropriate devices, viewers see one 2D regular image/frame with the left eye and the other 2D image/frame with the right eye. Most pixels in one image seen by one eye will have corresponding pixels in the other image seen by the other eye, except for occluded regions and regions near to frame boundaries. Let s be the disparity between a pair of corresponding pixels of a stereo image, e be the interaxial distance between the viewer's two eyes, and t be the distance between the viewer and the screen. According to Cormack and Fox (1985), the depth of the pixel pair that the viewer perceives, Z, can be determined as follows (see Fig. 1 for a simplified view configuration):

$$Z = \frac{et}{e - s}. \tag{1}$$

Here, we assume that the two 2D images of a stereo image are rectified such that the epipolar lines are aligned with the horizontal scanlines. As can be seen from Eq. 1, pixels in the image will be perceived to be behind the screen if they have uncrossed (positive) disparity, and in front of the screen if they have crossed (negative) disparity.

Today, almost all of the stereoscopic 3D movies are captured for playing in cinemas, which have very large screens and are located far from viewers. When we watch these

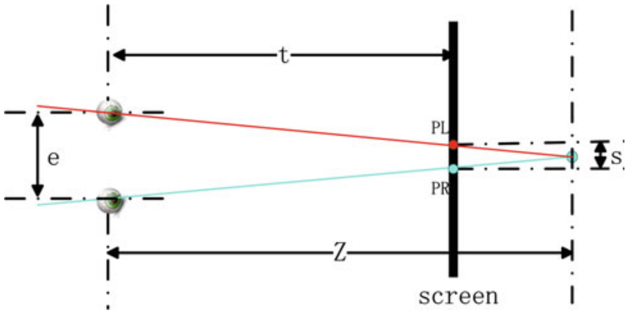


Fig. 1 A simplified 3D view configuration. *e* is the interaxial distance. *t* is the screen distance from the viewer. *s* is the distance between a pair of corresponding pixels, PL and PR. Z is the perceived pixel depth (Color figure online)

movies on a smaller screen, such as a TV or a computer screen, the 3D effects are much less obvious, due to the reducted depth range (Chauvier et al. 2010). In some extreme cases, viewers may hardly observe any 3D effects. On the other hand, if we watch a 3D movie on a larger screen, which was originally captured for small screens, the depth range can become too large and be out of the stereo comfort zone of the screen. In such a situation, viewers will feel uncomfortable and fatigue.

Consider a stereo image originally captured for a screen of pixel density β_1 located at distance t from the viewer. This image is now displayed on a screen of pixel density β_2 located at distance t' from the viewer. If we assume that $\lambda_t = t'/t$ and e does not change, the new disparity, s', is then: $s' = \lambda_s s$, where $\lambda_s = \beta_1/\beta_2$. The new depth of the pixel pair, Z', is:

$$Z' = \frac{e\lambda_t t}{e - \lambda_s s}. \tag{2}$$

If we assume that the screen size changes linearly with t, then $\lambda_s = \lambda_t$. The ratio between Eqs. 2 and 1 becomes:

$$\frac{Z'}{Z} = \frac{\lambda_t(e - s)}{e - \lambda_t s}. \tag{3}$$

Figure 2 shows a plot of the depth ratio between Z' and Z (i.e., Z'/Z) with respect to λ_t. We can see that as we reduce λ_t, the depth ratio drops non-linearly from 1 to roughly 0.85 (when $s = 0.9e$, which represents a more extreme disparity). However, the depth ratio becomes roughly linear from 0.15 to 0. As we play a 3D movie produced for large screens to a home TV or even handheld device, the value of λ_t is typically smaller than 0.1. Therefore, the depth ratio is much smaller than 0.1. Considering that a linear model is both simple and efficient, we therefore propose a linear depth mapping method in this paper.

The objective of this work is to develop an effective method for adapting the depth range of stereoscopic 3D videos. We propose to linearly scale the original depth range of stereoscopic videos to a new depth range according to the display setting, in order to preserve the 3D effects. The proposed method also minimizes distortion introduced to the videos due to the depth mapping process by preserving spatial features, such as lines and planes, and depth distances among neighboring features. Here, the *depth distance* refers to the difference in depth between two features in a 3D image as perceived by the viewer. In addition, unlike most previous methods that consider temporal coherence of the left and right videos separately, the proposed method considers both left and right images together to ensure depth coherence. It also considers object motion coherence in order to ensure stable 3D motion in the output videos. This is achieved by modeling the motion trajectories of keypoints and correcting the differences among the trajectories of neighboring keypoints across video frames.

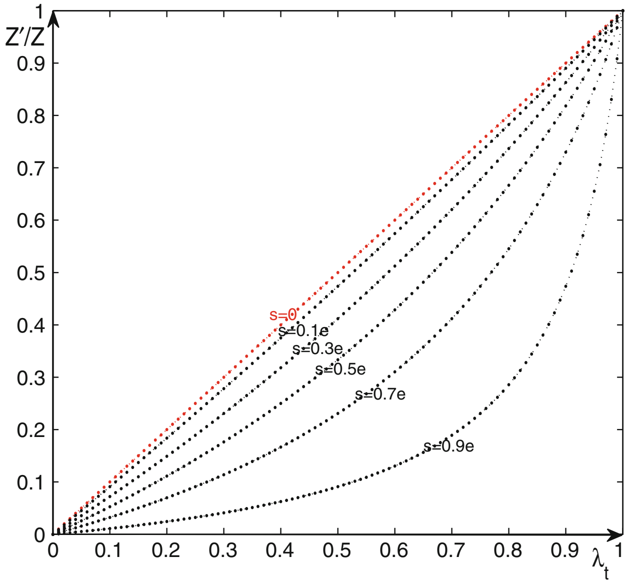


Fig. 2 Z'/Z changes with λ_t, where Z and Z' are defined by Eq. 1 and 2, respectively (Color figure online)

Our main contributions of this work can be summarized as follows:

- We propose a linear depth mapping algorithm to remap the depth range of 3D videos according to the actual display setting. The new depth range can also be adjusted according to the stereo comfort zone.
- We preserve spatial features across video frames by detecting and preserving lines and planes in the 3D videos. We also preserve relative depth distances among neighboring features.
- We enforce temporal coherence of depth and object motion across video frames by modeling the motion trajectories of keypoints and correcting the differences among the trajectories of neighboring keypoints in order to produce stable 3D motion across video frames.

The rest of this paper is organized as follows. Section 2 briefly summarizes related works. Section 3 gives an overview of our approach. Section 4 presents our depth mapping model and Section 5 discusses content preservation. Section 6 discusses depth and motion coherences to ensure temporal smoothness of the output videos. Section 7 presents some experimental results and user evaluations. Finally, Section 8 briefly concludes the work presented in this paper.

2 Related Work

3D content capturing, editing and displaying are attracting a lot of research interests in recent years due to the success of some 3D movies. Existing works on stereoscopic image/video depth/disparity adjustment mainly focus on changing image or feature disparity. We can classify them into two main types, one based on adjusting camera parameters during live 3D video capture and the other based on post-processing of captured 3D videos.

2.1 Depth Adjustment During Video Capture

Recently, there are several methods proposed that can be used to adjust the disparity/depth range of input images/videos by automatically adjusting the camera baseline and other parameters (Heinzle et al. 2011; Koppal et al. 2011).

To address the challenges in 3D production, Heinzle et al. (2011) present a novel design of a computational stereo camera system, which closes the control loop from capturing and analyzing stereoscopic videos to automatically adjusting some parameters of the stereo video capturing system, such as interaxial and convergence. They have developed intuitive interaction metaphors that automatically abstract and replace the cumbersome handling of rig parameters. Real-time performance and computational flexibility are enabled by the combination of FPGA, GPU, and CPU processing. However, this system can only be used for live 3D video capture, not for 3D video post-processing.

Koppal et al. (2011) build a viewer centric system that performs scene analysis and provides tools for shot planning. Similar to Heinzle et al. (2011), this system allows some basic correction and optimization of the stereo video content, but it can only be used for live 3D image/video capture, not for 3D video post-processing.

2.2 Depth Post-Processing

Some methods for post-processing of 3D images/videos are proposed (Chang et al. 2011; Guttmann et al. 2009; Lang et al. 2010; Lin et al. 2011; Liu et al. 2011; Wang et al. 2008a). Guttmann et al. (2009) propose an interactive method for users to dynamically input depth information to a 2D video. Based on the input depth information, an image warping method is used to convert the 2D video into a 3D video.

Lang et al. (2010) propose four simple operators for adjusting the disparity range of 3D videos: linear operator, nonlinear operator, gradient domain operator and temporal operator. While the linear and nonlinear operators are for editing of the depth range of individual frames, the gradient domain operator helps determine the weights of the linear and nonlinear operators in local regions of each frame. The temporal operator smooths scene transitions by adjusting the weights of the other operators across nearby frames. However, they do not consider relative depth changes among neighboring features. They also do not consider content preservation, such as planes, and left/right frame coherence. Although the paper mentions that line preservation is considered, we can actually see bended lines occurring very frequently in some of their results included in this paper. We suspect that this may be partly due to the fact that they do not consider left/right frame coherence.

Chang et al. (2011) propose an approach for resizing stereoscopic images without distorting the shape of prominent objects or reducing the consistency between left and right images. In situations where image aspect ratio changes but viewing configuration does not change, they attempt to keep the disparity of image features the same as that of the original. In situations where viewing configurations change, they argue that the disparity should be scaled accordingly and they utilize a linear disparity scaling operator for adjusting the disparity range. However, this work mainly focuses on 3D image display adaptation and linearly scaling the disparity range, without considering that simply scaling the disparity may introduce distortion to 3D image content.

Lin et al. (2011) propose some principles to reduce visual fatigue and enhance depth perception through cropping and warping. Liu et al. (2011) introduce some 3D cinematograph principles and illustrate their usage in stereoscopic image processing. They give two applications, video stabilization and photo slideshows, to show how these principles could be utilized in 3D media authoring. Wang et al. (2008a) propose a method for inpainting of stereo images by joining color and depth information. Liao et al. (2011) design a system to convert videos from 2D to 3D by combining motion analysis and user interaction. They propose two brushes, a depth difference brush and a depth equivalence brush, for users to edit the depth field of each frame. Smolic et al. (2011) survey the state-of-the-art in 3D video post-processing. Zilly et al. (2011) review some production rules needed for the acquisition of stereo content, and discuss a camera assistance system for stereo acquisition and production.

In conclusion, some existing methods are proposed to address the problems related to live 3D video capture. Other methods are proposed for post-processing 3D images/videos.

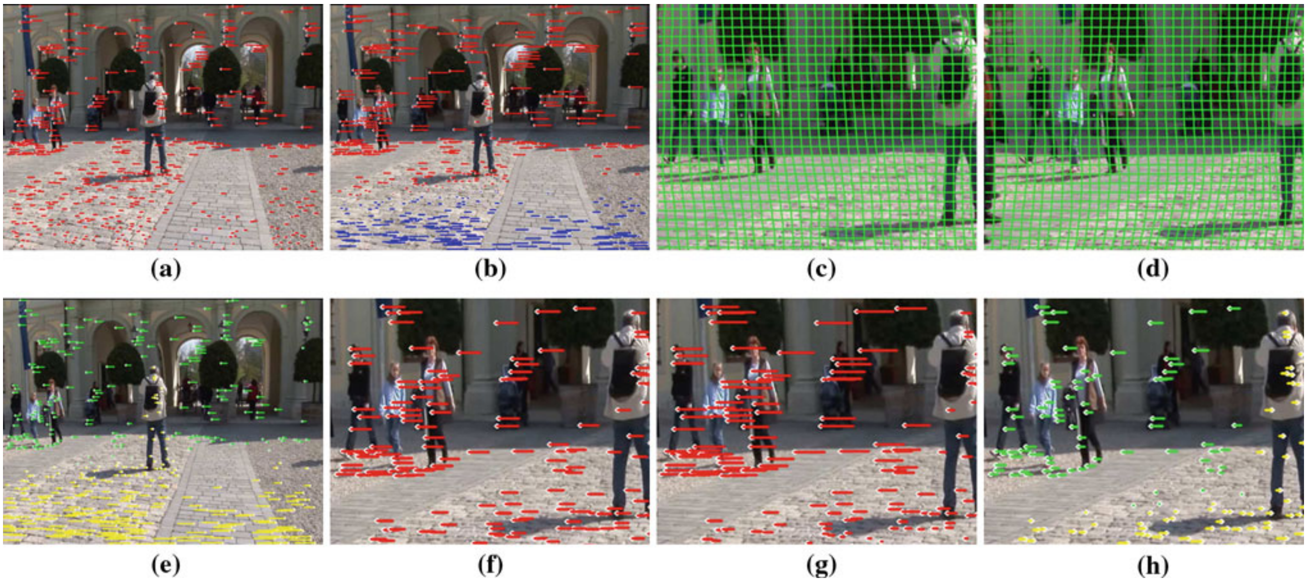


Fig. 3 Method overview: (**a**) matched keypoints and estimated disparities (*red* for positive and *blue* for negative) of the original left image; (**b**) matched keypoints and their disparities after depth mapping; (**c**) and (**d**) enlarged left and right images with warped meshes after depth mapping; (**e**) matched keypoint disparity differences between (**b**) and (**a**) (*green* for positive and *yellow* for negative); (**f**) enlarged version of (**a**); (**g**) enlarged version of (**b**); **h** enlarged version of (**e**) (Color figure online)

However, according to our study, most methods do not consider content preservation, nor do they consider the coherences between left/right images and across consecutive frames. In contrast, our work aims at producing a more complete system that considers depth remapping, feature preservation, left/right image coherence (or depth coherence) and content coherence (or motion coherence). As shown from our results, our method produces more stable and smooth outputs.

3 Overview of Our Approach

We remap the depth range of a stereo frame by adjusting the disparity of image features. To determine feature correspondences between the left and right frames for disparity adjustment, we use the method presented in Lowe (2004) to extract SIFT keypoints in the left and right frames separately. We then perform a matching process between the two sets of extracted keypoints to produce a set of matched keypoint pairs. Based on the feature tracking algorithm proposed by Bouguet (2000), we track the motion of these matched keypoint pairs across video frames. After obtaining the motion trajectories of these keypoint pairs, we smooth the trajectories to eliminate jittering. By modifying the disparity of the matched keypoint pairs, we can remap the original depth range of these keypoint pairs to the target depth range.

We then construct quad meshes in both left and right frames based on the extracted keypoints. By utilizing any appropriate image warping techniques for image retargeting (Wang et al. 2008b; Niu et al. 2010; Wang et al. 2011), we may adjust the disparity of 3D image features. We use the mean-value coordinate to represent the position of a keypoint in the left or right frame as $x = \sum w_i * v_i$, where v_i is one of the four vertices of the quad that x belongs to and w_i is the weight of this vertex. Since the disparity of a point corresponds to the difference between its horizontal coordinates in the left and right frames, we only need to modify the horizontal coordinates of the matched keypoints.

In order to incorporate user's high-level information for 3D video processing, we also allow users to optionally specify some features, such as interested objects in the left or right keyframe. Corresponding features can be automatically located in the other frame by matched keypoint fitting. Then, the specified constraints are automatically propagated to neighboring frames. Fig. 3 gives an overview of our method, from keypoint matching in Fig. 3a, to keypoint remapping in Fig. 3b, and mesh warping in Fig. 3c, d. Figure 3e shows the keypoint disparity differences before and after depth mapping.

To enforce stereoscopic 3D video depth and motion coherences, we map the depth of the 3D movie in three steps:

1. We separately map the original depth range of each 3D frame to a new depth range. (Refer to Eq. 20.)

2. We model the motion trajectories of keypoints across frames and correct the variation in differences among neighboring keypoint trajectories to enforce depth and motion coherences of the movie (Refer to Eq. 29.)
3. We map the depth range of each original 3D frame, again guided by the motion trajectories of keypoints obtained in step 2. (Refer to Eq. 30.)

To simplify our discussion, we adopt the following representations on disparity values and positions. A symbol, such as s or x, represents the original value (i.e., s represents the original disparity value of an image point before depth remapping). A symbol with $\hat{}$, such as $\hat{s}$ or $\hat{x}$, represents the ideal value after depth remapping (i.e., $\hat{s}$ represents the ideal disparity after depth remapping but before optimization). Finally, a symbol with $'$, such as s' or x', represents the actual value after depth remapping (i.e., s' represents the actual disparity after depth remapping and optimization).

4 Our Depth Mapping Model

In this work, we remap the original depth range of a 3D movie to a new depth range based on changing the image feature disparity. Let Z and $\hat{Z}$ be the depths of a point perceived by the viewer when watching a 3D movie on a target screen before and after the depth mapping process, respectively. We also let $[Z_{min}, Z_{max}]$ and $[\hat{Z}_{min}, \hat{Z}_{max}]$ be the original and the new depth ranges, respectively, of the 3D movie. Hence, a linear depth scaling model may scale the original depth Z to a new depth $\hat{Z}$ as follows:

$$\hat{Z} = K(Z - Z_{min}) + \hat{Z}_{min}, \qquad (4)$$

where

$$K = \frac{\hat{Z}_{max} - \hat{Z}_{min}}{Z_{max} - Z_{min}}.$$

Accommodation, convergence, and pupillary dynamics, i.e., the ocular near triad, continuously interact to control the functioning of the eyes (Takeda et al. 1999). Researchers working on visual discomfort and visual fatigue of stereoscopic displays have found that the depth range perceived by viewers in stereoscopic is determined by retinal disparity angle (Cormack and Fox 1985). For the stereo comfort zone, retinal disparity angle is limited to 1°. With long stimulus durations and convergence eye movements, retinal disparity can be larger and brought into fusion range without diplopia (Lambooij et al. 2009). Of course, comfortable depth range, viewer discomfort and fatigue are also related to the video content, such as motion and disparity changing (Lambooij and Heynderickx 2011). Thus, we supply users with an approach to remap the depth range of 3D videos to their favorite depth range. If we assume that the convergence point of the viewer's eyes is at the center of the stereoscopic image displayed on the screen and let η_1 and η_2 denote the negative and positive retinal disparity limits, respectively, then we may use these two limits to determine the target depth range as follows:

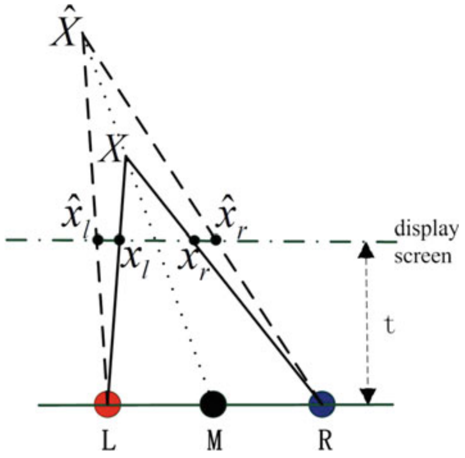


Fig. 4 Effect of depth mapping. L and R represent viewer's left and right eyes. M is the cyclopean eye located in the middle of L and R. M, X and $\hat{X}$ are on the same line. We can obtain relationship: $x_l + x_r = \hat{x}_l + \hat{x}_r$ (Color figure online)

$$\hat{Z}_{min} = \frac{et}{e - \eta_1 t} \quad \text{and} \quad \hat{Z}_{max} = \frac{et}{e - \eta_2 t} \qquad (5)$$

Our experimental results demonstrate that this approach to represent the target depth range is very effective.

By substituting $\hat{Z} = \frac{et}{e - \hat{s}}$ and Eq. 1 into Eq. 4, we may compute the ideal disparity $\hat{s}$ as follows:

$$\hat{s} = f(s) = \frac{(K-1)e^2 t + ets - e(e-s)(KZ_{min} - \hat{Z}_{min})}{Ket - (e-s)(KZ_{min} - \hat{Z}_{min})}. \qquad (6)$$

If the target depth range is enlarged linearly by K times with only depth scaling such that $\hat{Z}_{min} = KZ_{min}$, Eq. 6 can be simplified as

$$f(s) = \frac{(K-1)e + s}{K}. \qquad (7)$$

For our depth mapping approach (see Fig. 4), a matched keypoint pair should satisfy the relationship of $x_l + x_r = \hat{x}_l + \hat{x}_r$ after depth mapping. Hence, we define our depth mapping operator as follows:

$$\hat{x}_l = \frac{1}{2}(x_l + x_r - f(s)\beta) \quad \text{and} \quad \hat{x}_r = \frac{1}{2}(x_l + x_r + f(s)\beta). \qquad (8)$$

where β is the pixel density.

We may then apply the following energy terms in our depth range mapping of the left and right images:

$$E_{zl} = \sum_i \|x'_{i,l} - \hat{x}_{i,l}\|^2 \quad \text{and} \quad E_{zr} = \sum_i \|x'_{i,r} - \hat{x}_{i,r}\|^2. \qquad (9)$$

We combine the above two energy terms in our depth mapping process as follows:

$$E_z = E_{zl} + E_{zr}. \tag{10}$$

E_z can also be combined with other energy terms defined in Sect. 5 to preserve image features in each frame. By minimizing the total energy term, we can then obtain a new (or actual) disparity value of each matched keypoint pair and new positions for mesh vertices in the left and right images. At the same time, we also minimize distortions introduced to the image content due to the depth mapping process.

5 Stereoscopic Feature Preservation

As we remap the depth range of each pair of stereoscopic video frames, we not only change the disparity of each matched keypoint pair, but also preserve image features, such as relative depth distance between neighboring features, lines and plane surfaces. Our system can automatically incorporate the detected image features and any user specified high-level information in the depth mapping process.

5.1 Depth Preservation of Neighboring Features

The 3D depth field that a viewer perceives from a stereoscopic 3D image is mainly conveyed by the relative depths among neighboring objects/features in the 3D image. Hence, we need to preserve the relative depth distances among neighboring features in order to avoid cardboard effects after depth mapping (Ward et al. 2011). This will also help preserve the 3D scene structure. We achieve this with the following constraint:

$$E_{rel-z} = \sum_i \sum_{j \in N_i} \|(s'_i - s'_j) - (\hat{s}_i - \hat{s}_j)\|^2, \tag{11}$$

where N_i is the set of neighboring keypoints to i, $\hat{s}_i - \hat{s}_j$ is the ideal disparity difference between keypoints i and j after depth mapping, and $s'_i - s'_j$ is the actual disparity difference between keypoints i and j after depth mapping and optimization. In our implementation, we set the neighboring threshold as one-eighth of the image width. This means that all features that are within this threshold distance from keypoint i are considered as neighboring features of i. We utilize the energy term E_{rel-z} to preserve the change in relative depth distances among neighboring keypoints. Our results show that this constraint helps produce smoother object depths and prevent cardboard effects.

5.2 Mesh Edge Preservation

In low texture regions, objects may be stretched or squeezed after depth mapping due to the lack of matched keypoints in these regions. Preserving object shape in these regions is therefore important. We achieve this by preserving the length of mesh edges.

Let $x_{i,j}$ denote the horizontal coordinate of a mesh vertex in i row and j column. We introduce the following energy term for preserving horizontal length of mesh edges:

$$E_{length} = \sum_{i,j} \|(x'_{i,j+1} - x'_{i,j}) - (x_{i,j+1} - x_{i,j})\|^2. \tag{12}$$

We also try to preserve the linearity of vertical mesh edges to be in the same column as follows:

$$E_{align} = \sum_{i,j} \|2x'_{i,j} - x'_{i+1,j} - x'_{i-1,j}\|^2, \tag{13}$$

where $x_{i+1,j}$ and $x_{i-1,j}$ are the horizontal coordinates of mesh vertices above and below $x_{i,j}$ of column j. $x'_{i+1,j}$, $x'_{i-1,j}$ and $x'_{i,j}$ are the actual horizontal coordinate after depth mapping.

5.3 Line Preservation

Straight lines appearing in a movie often cross multiple quads constructed by our depth mapping algorithm. Consider a line l and refer to the sequence of mesh edges (both horizontal and vertical) that it crosses as $(x_1, y_1), (x_2, y_2)...(x_n, y_n)$. After depth mapping, their coordinates become $(x'_1, y'_1), (x'_2, y'_2)...(x'_n, y'_n)$. Since the vertical coordinate of each pixel does not change after depth mapping, which means $\frac{x'_{i-1}-x'_i}{y_{i-1}-y_i} = \frac{x'_i-x'_{i+1}}{y_i-y_{i+1}}$, we introduce the following energy term to prevent lines from bending:

$$E_{line} = \sum_l \sum_{i \in l} \|(x'_{i-1} - x'_i) - \frac{y_{i-1} - y_i}{y_i - y_{i+1}}(x'_i - x'_{i+1})\|^2. \tag{14}$$

Generally speaking, since points on a line may have different disparity values, a line projected to the left and to the right images may be rotated in opposite directions after depth mapping. Hence, we should allow lines to rotate.

This is different from image resizing, where line rotation should not be allowed. However, if a line is vertical and points on it are with same disparity, its orientation should be maintained after depth mapping. For example, if a pillar is vertical to the ground and points on it are of the same disparity, then it should still be vertical to the ground after depth mapping.

5.4 Plane Preservation

As our method is based on adjusting the disparity of matched keypoints and image warping, planes may be distorted after depth mapping. The main reason is that keypoints originally lie on a 3D plane may no longer lie on the same plane after depth mapping. We show our proof on this in the Appendix.

We address this problem by utilizing plane fitting in the original 3D image and then plane preservation in the depth mapping process. Let (x_l, y) and (x_r, y) be a pair of matched keypoints on a plane. If we fix our view point at $(\frac{w}{2\beta}, \frac{h}{2\beta}, 0)$, where h and w are the height and width of a 3D image, the coordinate of the matched keypoint pair in 3D space is:

$$\begin{aligned} X &= \frac{e}{e\beta - (x_r - x_l)}\left(\frac{x_r + x_l - w}{2}\right), \\ Y &= \frac{e}{e\beta - (x_r - x_l)}\left(y - \frac{h}{2}\right), \\ Z &= \frac{et\beta}{e\beta - (x_r - x_l)}. \end{aligned} \tag{15}$$

We extract 3D planes from the original 3D image and then identify matched keypoints that are on the same planes as follows:

1. We triangulate the keypoints on the original left (or right) image and compute the normal of each triangle.
2. If the normals of some adjacent triangles are similar, we combine these triangles to form a small plane and update its normal.
3. We further cluster small adjacent planes into larger ones and update their normals iteratively until no two plane clusters can be combined together.
4. Finally, if a plane cluster contains at least a certain number of keypoints (30 in our implementation), we output this as a plane.

After we have obtained the objective coordinates of keypoints in each frame, we fit a plane to a keypoint set that are originally on the same 3D plane. We use $D = a_l x + b_l y + c_l$ to present a plane in left image and $D = a_r x + b_r y + c_r$ to represent its corresponding plane in right image (see Appendix), where D is the disparity value. a_l, b_l, c_l, a_r, b_r, and c_r are parameters to be solved by a least square method.

In order to ensure that a matched keypoint pair, $(\hat{x}_{i,l}, y_i)$ and $(\hat{x}_{i,r}, y_i)$, lie on the target plane, which is separately mapped to the left and right images, we define the following energy terms:

$$\begin{aligned} E_{lp,i} = &\quad \|a_l\tilde{x}_{i,l} + b_l y_i + c_l - (\tilde{x}_{i,l} - \tilde{x}_{i,r})\|^2, \\ E_{rp,i} = &\quad \|a_r\tilde{x}_{i,r} + b_r y_i + c_r - (\tilde{x}_{i,l} - \tilde{x}_{i,r})\|^2, \end{aligned} \tag{16}$$

where $(\tilde{x}_{i,l}, y_i)$ and $(\tilde{x}_{i,r}, y_i)$ are the target positions of $(\hat{x}_{i,l}, y_i)$ and $(\hat{x}_{i,r}, y_i)$, respectively, after optimization.

In fact, a_l, a_r and b_l, b_r are always smaller than 0.01. If we simply solve Eq. 16 by combining it with constraint $\tilde{x}_{i,r} - \tilde{x}_{i,l} = \hat{x}_{i,l} - \hat{x}_{i,r}$, the objective horizontal coordinates of the matched keypoint pair may be shifted far away from their original position, due to the small coefficients. Hence, we introduce another constraint energy term as follows, which aims to prevent keypoints to be shifted too far away from their original positions:

$$E_{cp,i} = \|\tilde{x}_{i,l} - \hat{x}_{i,l}\|^2 + \|\tilde{x}_{i,r} - \hat{x}_{i,r}\|^2. \tag{17}$$

By minimizing the following energy term and assigning the computed values of $(\tilde{x}_{i,l}, y_i)$ and $(\tilde{x}_{i,r}, y_i)$ to $(\hat{x}_{i,l}, y_i)$ and $(\hat{x}_{i,r}, y_i)$, we may update the objective horizontal coordinates of a set of matched keypoint pairs that fall on the target plane:

$$(\tilde{x}_l, \tilde{x}_r) = \underset{\tilde{x}_l, \tilde{x}_r}{\operatorname{argmin}} \sum_{i \in h} (E_{lp,i} + E_{rp,i} + \omega E_{cp,i}), \tag{18}$$

where i is a keypoint on plane h, ω is empirically set as 1.0×10^{-5} in our experiment. Therefore, we can define our plane preservation energy term as

$$E_{plane} = \sum_{i} \sum_{i \in h} (\|x'_{i,l} - \hat{x}_{i,l}\|^2 + \|x'_{i,r} - \hat{x}_{i,r}\|^2). \tag{19}$$

In conclusion, for each frame in the depth mapping process, we compute the optimized coordinates of mesh vertices associated with the stereo frame by minimizing the following energy term:

$$\begin{aligned} E_{frame} = &\ E_z + w_2 E_{rel-z} + w_3 E_{length} + w_4 E_{align} \\ &+ w_5 E_{line} + w_6 E_{plane}. \end{aligned} \tag{20}$$

In our experiments, we set the parameter values as follows: $w_2 = 2.0$, $w_3 = 1.0$, $w_4 = 1.0$, $w_5 = 1000$ and $w_6 = 100$.

6 Depth Coherence and Motion Coherence

In this section, we consider two types of coherence, depth coherence and motion coherence. Our main concern here is to enforce smooth changes among neighboring features, in terms of depth and position, in order to preserve the smoothless of the depth field and the image content across frames. Our depth coherence constraint is to ensure that the depth difference between any two neighboring keypoints changes smoothly over time to ensure that the depth field of the video changes smoothly. Our motion coherence constraint is to ensure that the position difference between any two neighboring keypoints changes smoothly over time to ensure that the image content of the video changes smoothly.

To achieve depth and motion coherences, we model the motion trajectories of keypoints across frames and try to minimize the variation of the differences among neighboring keypoints across frames. Here, our assumption is that after the depth mapping process, the motion trajectories and the depth trajectories of keypoints should both be smooth. Note that we consider two keypoints as neighboring keypoints if both appear in a continuous frame sequence simultaneously and the distance between them is less than a specified threshold.

As the vertical position of a keypoint pair does not change after depth mapping, we simply model the motion (or depth) trajectory of a keypoint i as 1D (i.e., horizontal only) scaling plus translation. Hence, given the positions of keypoint i between frames m to n as $P_i = p_i^m, p_i^{m+1}, \ldots, p_i^n$, the motion trajectory of i can be written as $\hat{p}_i = a_i p_i + b_i$.

6.1 Depth Coherence

Consider two neighboring keypoints i and j of a stereo image with disparity s_i and s_j, respectively. After depth mapping, the new depth difference between them is scaled by $k_{i,j}$ times in the depth remapped stereo image, and can be represented as

$$\frac{et}{e-\hat{s}_i} - \frac{et}{e-\hat{s}_j} = k_{i,j}\left(\frac{et}{e-s_i} - \frac{et}{e-s_j}\right). \tag{21}$$

From Eq. 21, we can obtain the relationship between $\hat{s}_i - \hat{s}_j$ and $s_i - s_j$ as follows:

$$\hat{s}_i - \hat{s}_j = C_{i,j}(s_i - s_j), \tag{22}$$

where

$$C_{i,j} = k_{i,j}\frac{(e-\hat{s}_i)(e-\hat{s}_j)}{(e-s_i)(e-s_j)}. \tag{23}$$

As the disparity difference of the two neighboring keypoints, $\hat{s}_i - \hat{s}_j$, typically changes only slightly across a short sequence of frames, $C_{i,j}$ is assumed to be a constant. From Eq. 22, we may then say that the relationship between $\hat{s}_i - \hat{s}_j$ and $s_i - s_j$ is a linearly scaling. Consequently, we define an energy term for our depth coherence preservation across video frames as follows:

$$E_{coh-z} = \sum_{i,j}\sum_{t=m}^{n} \|((s'_{i,t} - s'_{j,t}) - C_{i,j}(s_{i,t} - s_{j,t})\|^2. \tag{24}$$

We may also represent Eq. 24 using the horizontal positions of matched keypoints in the left and right images as follows:

$$E_{coh-z} = \sum_{i,j}\sum_{t=m}^{n} \|((x'_{i,r,t} - x'_{i,l,t}) - (x'_{j,r,t} - x'_{j,l,t}) \\ - c'_{i,j}((x_{i,r,t} - x_{i,l,t}) - (x_{j,r,t} - x_{j,l,t}))\|^2, \tag{25}$$

where $x_{i,r} - x_{i,l}$ and $x_{j,r} - x_{j,l}$ are the original disparities of keypoints i and j in pixel units. $x'_{i,r} - x'_{i,l}$ and $x'_{j,r} - x'_{j,l}$ are their corresponding disparities after depth mapping, also in pixel units. $c'_{i,j}$ is the scaled disparity difference between the trajectories of keypoints i and j in 3D space. The objective of energy term E_{coh-z} is to enforce that the depth difference of keypoints i and j after depth mapping is similar to that before depth mapping across video frames.

6.2 Motion Coherence

We preserve the coherence of the left and right videos using two criteria. First, we assume that the motion trajectory of each keypoint across frames is smooth and we define our energy term as follows:

$$E_{coh-sm} = \sum_{i}\sum_{t=m}^{n-1} \\ (\|x'_{i,l,t} - x'_{i,l,t+1}\|^2 + \|x'_{i,r,t} - x'_{i,r,t+1}\|^2) \tag{26}$$

Second, we assume that the horizontal position difference between two neighboring keypoints has undergone a linear scaling through the depth mapping operation and we define our energy terms to preserve the coherence of the left video and the right video as follows:

$$\begin{aligned} E_{ml} &= \sum_{i,j}\sum_{t=m}^{n} \|(x'_{i,l,t} - x'_{j,l,t}) \\ &\quad - c'_{i,j,l}(x_{i,l,t} - x_{j,l,t})\|^2, \\ E_{mr} &= \sum_{i,j}\sum_{t=m}^{n} \|(x'_{i,r,t} - x'_{j,r,t}) \\ &\quad - c'_{i,j,r}(x_{i,r,t} - x_{j,r,t})\|^2, \end{aligned} \tag{27}$$

where $c'_{i,j,l}$ and $c'_{i,j,r}$ are constant. Note that $c'_{i,j,l}$ and $c'_{i,j,r}$ are independent of $c'_{i,j}$. The trajectories parameters $a_{i,l}$,$a_{i,r}$,$b_{i,l}$,$b_{i,r}$ for a matched keypoint pair i and $c'_{i,j}$, $c'_{i,j,l}$, $c'_{i,j,r}$ for two neighboring keypoints i and j, are all determined by our optimization process.

6.3 The Total Energy Term

After the depth mapping process of Eq. 20, we obtain the horizontal coordinates of each keypoint, i, for a sequence of consecutive frames as $Q_i = q_{i,m}, q_{i,m+1}, \cdots, q_{i,n}$, where m and n are frame numbers. We define an energy term to optimize the results from Eq. 20 by comparing each Q_i with its corresponding optimized trajectory as follows:

$$E_{traj} = \sum_{i}\sum_{t=m}^{n} \|a_i x_{i,t} + b_i - q_{i,t}\|^2. \tag{28}$$

We minimize the following energy terms to obtain optimized motion trajectory parameters for all matching keypoint pairs,

$$\begin{aligned} E_{coh} &= E_{coh-z} + \mu_1 E_{coh-sm} + \mu_2(E_{ml} + E_{mr}) \\ &\quad + \mu_3 E_{traj}, \end{aligned} \tag{29}$$

where $\mu_1 = 10$, $\mu_2 = 1.0$ and $\mu_3 = 1.0$.

Finally, we map the depth range of each original stereo frame again by adding the following energy term to Eq. 20 with a weight $w_7 = 1000$:

$$E_{remap} = E_{frame} + w_7 \sum_i \|x_i' - (a_i x_i + b_i)\|^2. \quad (30)$$

7 Experimental Results and Discussion

Our quadratic energy minimization problem is actually a least square regression problem that can be solved linearly. Since matrix decomposition and inversion have high computation costs for larger matrices, we only use Cholesky decomposition and symmetric positive definite matrix inversion for the first frame in each video clip in order to obtain an accurate initial result. Then, for each frame, we use the result of the previous frame as the initial value of each frame in an iterative process based on the conjugate gradient method. When running on a PC with an Intel i3 2.4GHz CPU and 4GB RAM, the speed of our method is about 1 second for each stereoscopic frame at a resolution of 600 × 800.

In Fig. 5a, we show that if an original stereo image produced for large screen is shown as a small image, its depth range may be reduced significantly. Using our depth mapping method, we can remap its depth range such that the disparity of the stereo image becomes visible as shown in Fig. 5b. Due to our feature preservation algorithms, we can see that the orientation of the bridge, the fence on the bridge and the buildings on the other side of the bridge appear natural. We do not notice any visible distortions. In Fig. 6, our method can effectively remap the original depth range to different depth ranges. Figure 6b, c remap the depth ranges so that they cover regions in front of and behind the screen, while Fig. 6d simply enlarges the depth range. We can see that image features, such as walls on the building, the stone road and the floor, are all well-preserved. Figure 7 shows another example of enlarging the depth range. As we increase the depth range here, we do not observe any cardboard effect in the resulting images.

Figure 8 remaps the depth range of a stereoscopic video clip of a moving train. Figure 8a–d show four selected frames of the clip. For each stereo pair of original frames shown at the top row, we scale up the original depth range to a target depth range by setting $\eta_1 = 0$ and $\eta_2 = 3°$. As compared with the original stereo frames, we can see that background objects, such as the hill and the trees, now appear further away in the output stereo frames shown at the bottom row. Relative depth distances among the background objects are

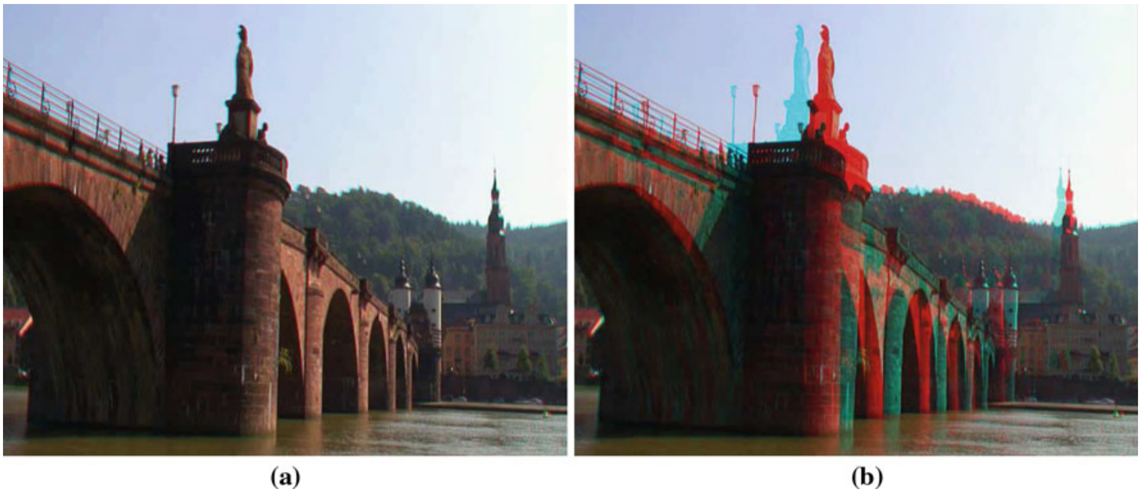


Fig. 5 (**a**) An original stereoscopic frame; (**b**) the output frame after our depth mapping process (Color figure online)

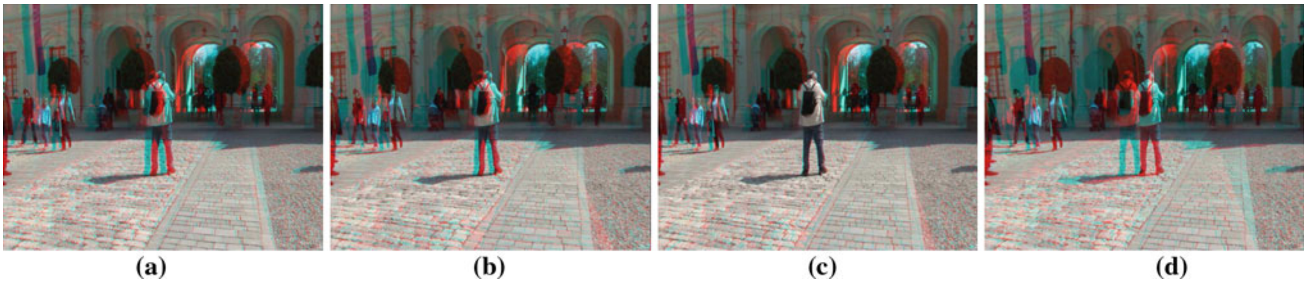


Fig. 6 Changing the target depth range by changing the negative and positive disparity limits: (**a**) an original stereoscopic frame; (**b**) target depth range set using $\eta_1 = -2°$ and $\eta_2 = 2°$; (**c**) target depth range set using $\eta_1 = -1°$ and $\eta_2 = 1°$; (**d**) target depth range set using $\eta_1 = 0$ and $\eta_2 = 3°$. Frame resolution is 648 × 840 (Color figure online)

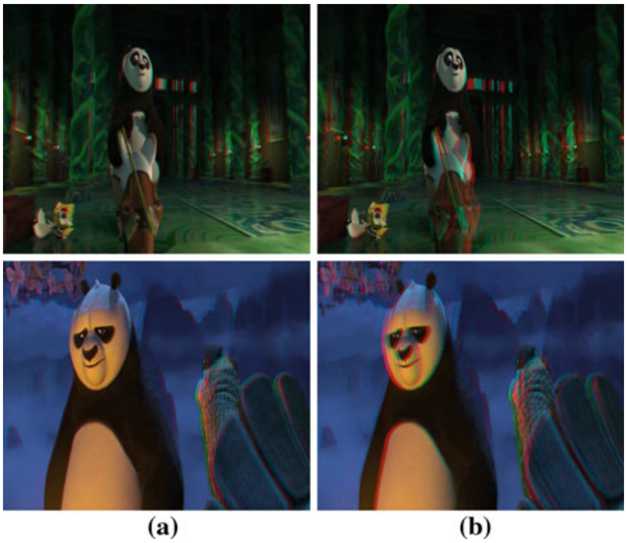


Fig. 7 Comparison between our results and original frames: (**a**) original stereoscopic frames; (**b**) our results with the target depth range set using $\eta_1 = -1°$ and $\eta_2 = 1°$. Frame size is 624 × 768 (Color figure online)

also increased. We can also see that regular objects, such as the train and the track, are preserved well without any obvious bending, not only between stereo pairs of output frames, but also across video frames (motion coherence). Likewise, depth changes across frames are also consistent (depth coherence).

We have also compared our method with Lang et al. (2010). Figure 9a shows two original stereo images. We double the original depth range shown in Fig. 9a to produce Fig. 9b using our method and Fig. 9c using Lang et al. (2010). From the upper images, we can see that the large stone floor in Fig. 9c appears uneven after depth mapping. In addition, the windows on left hand side of the building are slightly rotated in the left image. Finally, the depth distance between the person in the middle and the building appear similar to Fig. 9a, although we would expect them to get further apart as we increase the depth range. On the contrary, Fig. 9b does not have these problems. From the lower images, we may observe similar problems. We can see that the large stone floor in Fig. 9c again appears somewhat uneven. We can also observe different amounts of rotation between the left and right images (an obvious example is the upper second window from left end of the image). In addition, the black vertical pipe and pillar on the left end of the image are clearly distorted. Again, Fig. 9b does not have these problems.

Figure 10 further compares the differences between our result with Lang et al. (2010)'s result using the top example in Fig. 9. Figure 10a shows the original left image. Both Fig. 10b, c shows the depth remapping result from our method. In Fig. 10b, we also show the keypoint positions from our method (red points) and from Lang et al. (2010) (cyan points) for comparison, while in Fig. 10c, we show the keypoint disparity differences between our method and Lang et al. (2010) (with green lines representing positive disparity differences and yellow lines representing negative disparity differences, when subtracting their disparity values from ours). We can see that the differences appear almost everywhere in the image. Although most of the lines seem to be short, when the image is displayed on a large screen, the differences become significant.

Figure 11 shows another comparison. Again, we double the original depth range shown in Fig. 11a. Here, an obvious problem is that the relative depth distances among the buildings in Fig. 11c are rather similar to those in Fig. 11a.

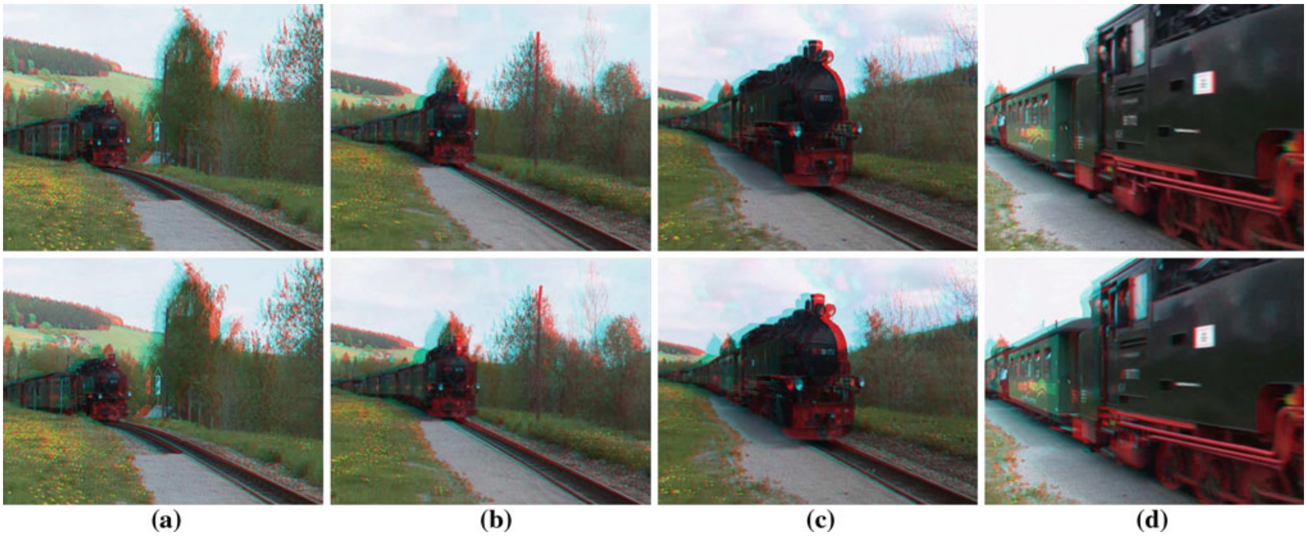


Fig. 8 Comparison between our results and an original frame sequence. *First row* shows the original frames. *Second row* shows our results with target depth range set using $\eta_1 = 0$ and $\eta_2 = 3°$. (**a**) the 0th frame; (**b**) the 200th frame; (**c**) the 300th frame; (**d**) the 350th frame. Frame size is 648 × 840 (Color figure online)

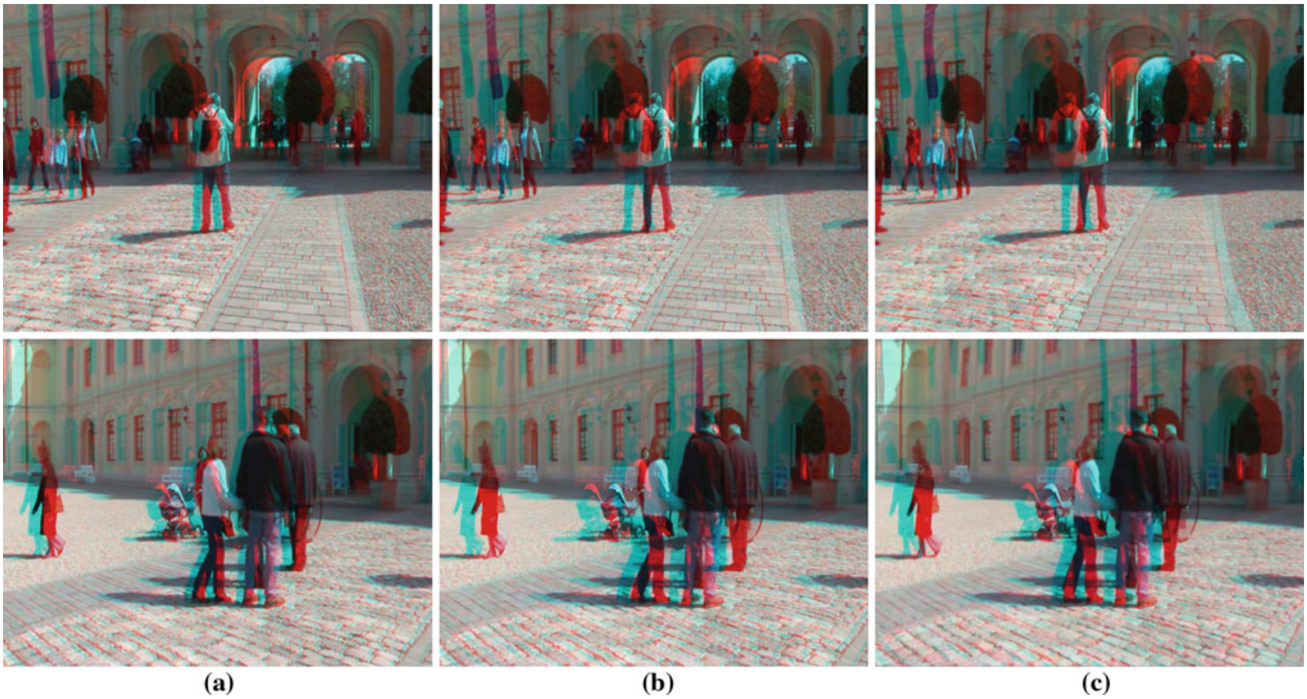


Fig. 9 Comparison between our results and Lang et al. (2010)'s results: (**a**) original stereoscopic frames; (**b**) our results; (**c**) Lang et al. (2010)'s results. (**b**) and (**c**) double the disparity range of (**a**). Frame size is 540 × 720 (Color figure online)

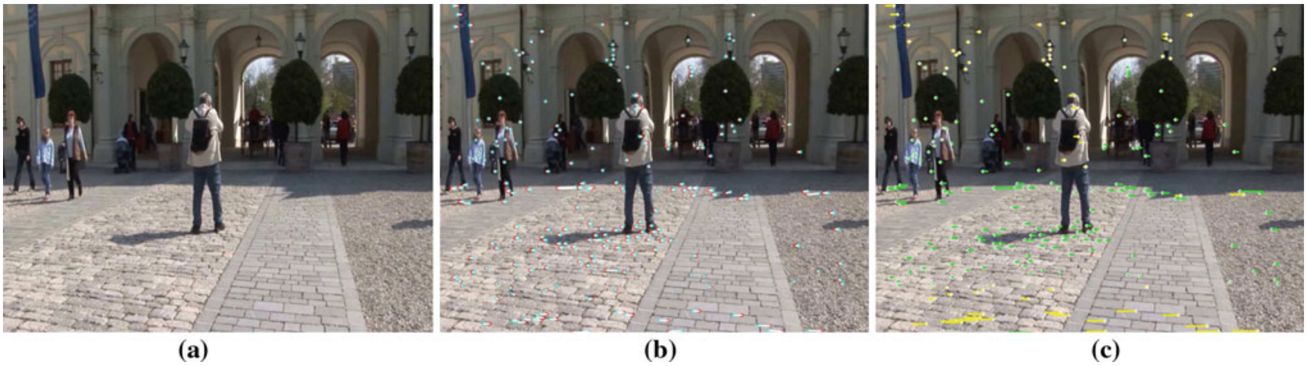


Fig. 10 Comparison between our result and Lang et al. (2010)'s result (refer to the top example in Fig. 9): (**a**) the original left image; (**b**) our result with keypoint positions (*red* and *cyan* points representing keypoint positions from our method and from Lang et al. (2010), respectively); (**c**) our result with keypoint disparity differences between our method and Lang et al. (2010) (*green* for positive disparity and *yellow* for negative disparity) (Color figure online)

We may also observe a small rotation in the status located in the middle of Fig. 11c (in particular in the lower image). On the other hand, the depth distances among the buildings are much more obvious in Fig. 11b and we do not observe any rotation in the status.

7.1 User Study

To further evaluate the effectiveness of our method, we have conducted a user study with 14 test users, who are students from various departments in USTC. These students are not familiar with this project. After we have completed all the experiments, we find that 2 of the test users do not have normal stereoacuity. As a result, we have to disregard the corresponding data, and our user study is then based on the data from the remaining 12 test users. We have selected a total of 17 3D test videos for our experiments. These test videos are mostly well-known 3D videos/clips. They contain rich information, such as people, vehicles, building, and objects with slow and fast motions. From these 17 test videos, we first produce a new set of 17 videos using our method. We also sent the first author of Lang et al. (2010) three of our test videos to produce another set of three videos using their method. We have conducted two

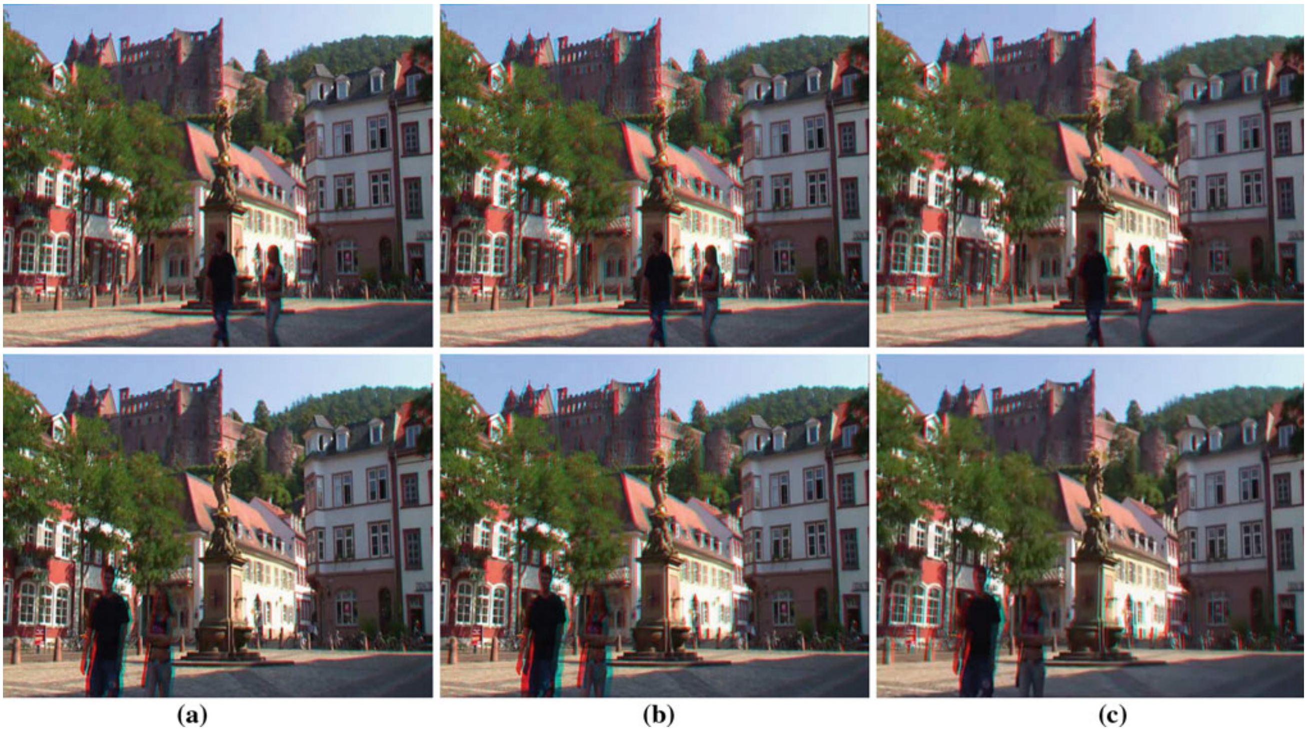


Fig. 11 Comparison between our results and Lang et al. (2010)'s results: (**a**) original stereoscopic frames; (**b**) our results; (**c**) Lang et al. (2010)'s results. (**b**) and (**c**) double the disparity range of (**a**). Frame size is 576 × 720 (Color figure online)

experiments based on these three sets of videos, i.e., original (14 videos), our method (14 + 3 videos) and Lang et al. (2010) (3 videos).

In both experiments, we use a 19-inch LCD screen of resolution 1280 × 1024 and pixel density $\beta = 3.413 pixels/mm$. We set $t = 500mm$. All the stereoscopic videos that we showed to our test users are represented as red(left)-cyan(right) anaglyph image sequences.

In our user study, our aims are to assess the methods on stereoscopic effects, video content preservation, temporal coherence and comfortability. We design five questions to ask the test users after each experiment as follows:

1. Rate the stereoscopic effects of the video (0–5):
2. Rate the observed distortion level of the video (0–5):
3. Rate the observed content jittering level of the video (0–5):
4. Rate the uncomfortability level of watching the video (0–5):
5. Which of the two videos would you prefer (left/right/both):

The first four questions concern about the quality of the videos. For the first question, the higher the score the better. For the other three questions, the lower the score the better. The fifth question is the user's overall rating.

Our first experiment is to compare the videos produced by our method with the original videos to see if our method would enhance the original videos. We showed the original videos on the left and ours on the right, but without telling the test users what they were. For the first four questions, the test users needed to give a score on each of the 28 videos (14 originals + 14 ours). For the fifth question, the test users needed to indicate which video that they prefer after watching each pair of the 14 video pairs. They could choose left, right or both.

Our second experiment is to compare the videos produced by Lang et al. (2010) with those by our method. In order to make it easier for the test users to compare the quality of the videos, we double the disparity range of the remaining three original videos using our method. We have also obtained three corresponding videos from the first author of Lang et al. (2010), also with the disparity range doubled. Hence, our experiment is based on these three pairs of videos. We conducted this experiment in a similar way as in the first experiment and the test users answered the same five questions in exactly the same way.

Figure 12 summarizes the results from Experiments 1 and 2. Figure 12a shows how the test users feel about different quality factors, including stereoscopic effect (Q1), distortion (Q2), jittering (Q3) and uncomfortability (Q4). The upper diagram shows the results of our method, while the lower diagram shows the results of the original videos (for Videos

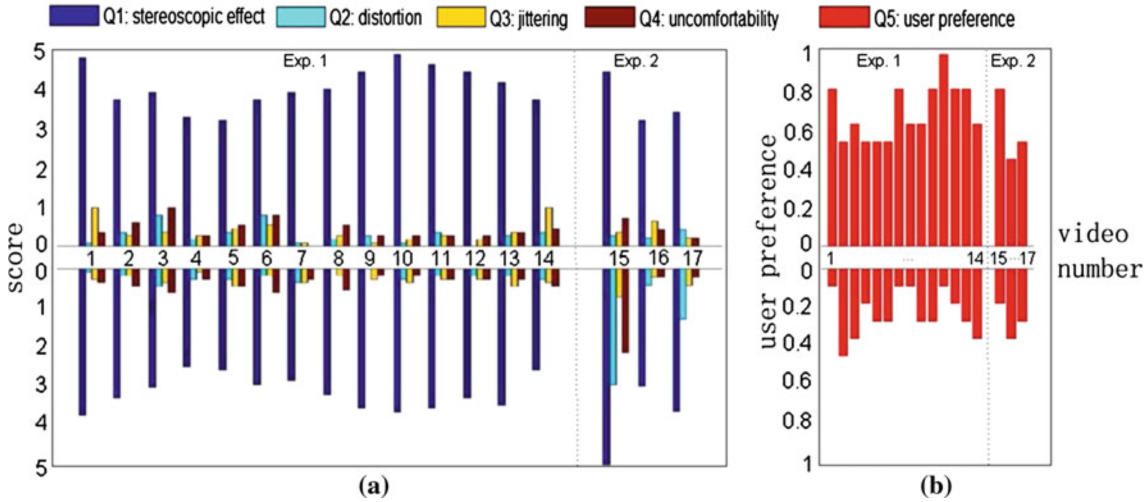


Fig. 12 Summary of our user study: (**a**) shows the average user scores relating to depth mapping quality (questions 1–4); (**b**) shows the overall user preference (question 5). Videos 1–14 are used in Experiment 1 to compare between our results and the original videos. Videos 15–17 are used in Experiment 2 to compare between our results and Lang et al. (2010)'s results. Upper parts of both (**a**) and (**b**) show the scores on our results, while the lower part of (**a**) shows the scores on the original videos and the lower part of (**b**) shows the scores on Lang et al. (2010)'s results (Color figure online)

1–14) and of Lang et al. (2010) (for Videos 15–17). Likewise, Fig. 12b shows the test users' overall preferences (Q5) on the videos.

From Experiment 1 of Fig. 12a, we can see that our method produces obviously better stereoscopic effects in almost all 14 videos when compared with the original videos. We can also see that there our videos have slightly higher average levels of observed distortion, jittering and uncomfortability compared to the original videos. (Here, it may be interesting to see that some test users also indicated that they observed distortions, etc. in the original videos. This may indicate that there is a small amount of distortion even in the original videos. It may also indicate that the two sets of videos are very similar in quality that some test users are not sure which are better.)

From Experiment 2 of Fig. 12a, we can see that although the test users observe similar levels stereoscopic effects from the two sets of videos, Lang et al. (2010)'s videos are given slightly higher scores. However, the test users generally observe a much higher levels of distortions, jittering and uncomfortability from Lang et al. (2010)'s videos, in particular from Video 15. Figures 9 and 11 show snapshots of Videos 15 and 17, respectively.

Figure 12b shows that majority of the test users prefer our videos than the original videos (in Experiment 1) and Lang et al. (2010)'s videos (in Experiment 2). According to the collected figures, in Experiment 1, 64.29 % of test users prefer our videos. 14.29 % of test users cannot distinguish which videos are better. Only 21.43 % of test users prefer the original videos. In Experiment 2, 55.56 % of test users prefer our videos. 19.44 % of test users cannot distinguish which videos are better. Only 25 % of test users prefer Lang et al. (2010)'s videos. Judging from these results, it seems that when the quality of the videos are nearly the same as in Experiment 1, most users would prefer one with better stereoscopic effects as shown in Fig. 12b: Experiment 1. However, when the quality of the videos are different as in Experiment 2, most users would prefer one with higher quality as shown in Fig. 12b: Experiment 2, even though Lang et al. (2010)'s videos have slightly better stereoscopic effects.

7.2 Limitations

Our depth mapping method uses image warping to simultaneously adjust the positions of image pixels in the left and right images in order to remap the depth range of a stereo image. Thus, it shares the same limitation as the other methods that utilize image warping. In some extreme situations, such as mapping a small depth range to a very large depth range, it may not be able to preserve all image features well. For example, if a scene contains a number of similar objects, these objects may be warped differently and their shapes may not be exactly the same after depth mapping. In addition, in some homogeneous regions, we may not be able to extract enough keypoints for proper warping.

8 Conclusion

In this paper we have presented a depth mapping method for stereoscopic video depth range mapping, which can effectively map the original depth range to the objective depth range according to the viewing configuration. The proposed method considers image content, depth coherence and motion coherence of the input stereoscopic video through our optimization steps. As a result of these optimization steps, our method improves the depth range of the stereoscopic videos, while at the same time prevents 3D scene structure from distortion. It also preserves 3D image features, such as relative depth distance among neighboring features. To enforce depth and temporal coherence across video frames, we model the motion trajectories of matched keypoint pairs and correct the variation in differences among neighboring keypoint trajectories, and then use them to guide our depth mapping process. Our experimental results demonstrate the effectiveness of the proposed method.

However, our current implementation of this method is too slow for real-time depth mapping, in particular for high-resolution videos. As a future work, we are currently investigating the possibility of a GPU implementation of our method. We believe that a real-time depth mapping algorithm will be extremely useful in streaming stereoscopic videos for playback on different devices.

For reference, we have placed some of the videos and additional results in a project webpage at: http://www.cs.cityu.edu.hk/~rynson/projects/3D/3D.html

Acknowledgments We would like to thank Qingxiong Yang for some useful suggestions and some insightful discussions on issues related to our work, and Antoni Chan for some comments. We would also like to thank Manuel Lang for helping produce some test videos using their system. The work presented in this paper was partially supported by a GRF grant from the Research Grants Council of Hong Kong (RGC Reference Number: CityU 116010) and a SRG grant from City University of Hong Kong (Project Number: 7002768).

Appendix

Let the calibration matrix of a CCD camera be:

$$K = \begin{bmatrix} f_x & s & t_x \\ 0 & f_y & t_y \\ 0 & 0 & 1 \end{bmatrix},$$

where f_x and f_y are the focal length of the camera (in terms of pixels) in the x and y directions, respectively. s is the skew parameter and $[t_x, t_y]$ is the principle point. Then we have:

$$K^{-1} = \begin{bmatrix} f_x^{-1} & -\frac{s}{f_x f_y} & -\frac{t_x}{f_x} - \frac{st_y}{f_x f_y} \\ 0 & f_y^{-1} & -\frac{t_y}{f_y} \\ 0 & 0 & 1 \end{bmatrix}.$$

Using K^{-1}, we can back project a 2D pixel located at $[\mathbf{x}, \mathbf{y}]$ into the 3D space with the knowledge of its depth value, Z. If we let the corresponding 3D point be $P = [X, Y, Z]^T$, then we have:

$$\begin{aligned}[X, Y, Z]^T &= Z \cdot K^{-1}[\mathbf{x}, \mathbf{y}, 1]^T \\ &= Z \begin{bmatrix} \frac{1}{f_x}\mathbf{x} - \frac{s}{f_x f_y}\mathbf{y} + (-\frac{t_x}{f_x} - \frac{st_y}{f_x f_y}) \\ \frac{\mathbf{y}}{f_y} - \frac{t_y}{f_y} \\ 1 \end{bmatrix}. \end{aligned} \quad (31)$$

If P lies on a 3D plane $\pi = [u, v, w, 1]^T$, then

$$\pi^T [P^T, 1]^T = [u, v, w, 1][X, Y, Z, 1]^T = 0. \quad (32)$$

Let the stereo baseline be B. The disparity value of pixel $[\mathbf{x}, \mathbf{y}]$ is then:

$$\begin{aligned} D &= \frac{f_x B}{Z} \\ &= (-uB)\mathbf{x} + \frac{B}{f_y}(us - vf_x)\mathbf{y} + u\left(\frac{t_x}{f_x} + \frac{st_y}{f_y} + \frac{vt_y}{f_y} - w\right). \end{aligned} \quad (33)$$

Let also the intrinsic parameters of the CCD camera be:

$$\begin{aligned} \hat{a} &= -uB, \\ \hat{b} &= \frac{B}{f_y}(us - vf_x), \\ \hat{c} &= u\left(\frac{t_x}{f_x} + \frac{st_y}{f_y} + \frac{vt_y}{f_y} - w\right), \end{aligned} \quad (34)$$

Given a 3D plane π, we can then rewrite Eq. 33 as

$$D = \hat{a}\mathbf{x} + \hat{b}\mathbf{y} + \hat{c}. \quad (35)$$

If we substitute Eq. 7 into Eq. 35 and set $x = x_l$, then we may obtain:

$$\begin{aligned} D' &= \hat{a}(\mathbf{x} - \frac{(K-1)(e-s)}{2K\beta}) + \hat{b}\mathbf{y} + \hat{c} \\ &= \frac{(K+1)\hat{a}}{2K}x + \hat{b}\mathbf{y} + \frac{(K-1)\hat{a}}{2K}x_r + \hat{c} + \frac{(K-1)e}{2K\beta}. \end{aligned} \quad (36)$$

Hence, after linear depth mapping, points originally lying on a plane may no longer be on the same plane.

References

Bouguet, J. (2000). *Pyramidal implementation of the lucas kanade feature tracker description of the algorithm*. Technical report: Intel Corporation Microprocessor Research Labs.

Chang, C., Liang, C., & Chuang, Y. (2011). Content-aware display adaptation and interactive editing for stereoscopic images. *IEEE Transactions on Multimedia*, *13*(4), 589–601.

Chauvier, L., Murray, K., Parnall, S., Taylor, R., & Walker, J. (2010) Does size matter? The impact of screen size on stereoscopic 3dtv. In *Presented at IBC conference* (www.visionik.dk/pdfs/3DTVDoesSizeMatter_IBC2010Award.pdf).

Cormack, R., & Fox, R. (1985). The computation of disparity and depth in stereograms. *Attention, Perception, and Psychophysics*, *38*, 375–380.

Guttmann, M., Wolf, L., & Cohen-Or, D. (2009). Semi-automatic stereo extraction from video footage. In *Proceedings of IEEE ICCV* (pp. 136–142). Kyoto.

Heinzle, S., Greisen, P., Gallup, D., Chen, C., Saner, D., Smolic, A., Burg, A., Matusik, W., & Gross, M. (2011). Computational stereo camera system with programmable control loop. *ACM Transactions on Graphics, 30*, 1–10.

Koppal, J., Zitnick, C., & Cohen, M. (2011). A viewer-centric editor for 3D movies. *IEEE Computer Graphics & Applications*, *31*, 20–35.

Lambooij, M., IJsselsteijn, W., Fortuin, M., & Heynderickx, I. (2009). Visual discomfort and visual fatigue of stereoscopic displays: A review. *Journal of Imaging Science and Technology, 53*, 30201–30214.

Lambooij, M., & Heynderickx, I. (2011). Visual discomfort of 3D tv: Assessment methods and modeling. *Displays*, *32*, 209–218.

Lang, M., Hornung, A., Wang, O., Poulakos, S., Smolic, A., & Gross, M. (2010). Nonlinear disparity mapping for stereoscopic 3D. *ACM Transaction on Graphics, 29*(3), 751–760.

Liao, M., Gao, J., Yang, R., & Gong, M. (2011). Video stereolization: Combining motion analysis with user interaction. *IEEE Transactions on Visualization and Computer Graphics 17*(12), 2035–2044.

Lin, H., Guan, S., Lee, C., & Ouhyoung, M. (2011). Stereoscopic 3D experience optimization using cropping and warping. In *ACM SIGGRAPH Asia Sketches*. Hong Kong.

Liu, C., Huang, T., Chang, M., Lee, K., Liang, C., & Chuang, Y. (2011). 3D cinematography principles and their applications to stereoscopic media processing. In *Proceedings of ACM multimedia* (pp. 253–262). Singapore.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*, 91–110.

Niu, Y., Liu, F., Li, X., & Gleicher, M. (2010). Warp propagation for video resizing. In *Proceedings of IEEE CVPR* (pp. 537–544). San Francisco.

Smolic, A., Kauff, P., Knorr, S., & Hornung, A. (2011). Kunter: Three-dimensional video postproduction and processing. *Proceedings of the IEEE*, *99*(4), 607–625.

Takeda, T., Hashimoto, K., Hiruma, N., & Fukui, Y. (1999). Characteristics of accommodation toward apparent depth. *Vision Research*, *39*, 2087–2097.

Wang, L., Jin, H., Yang, R., & Gong, M. (2008a). Stereoscopic inpainting: Joint color and depth completion from stereo images. In *Proceedings of IEEE CVPR* (pp. 1–8). Anchorage.

Wang, Y., Hsiao, J., Sorkine, O., & Lee, T. (2011). Scalable and coherent video resizing with per-frame optimization. *ACM Transactions on Graphics 30*. doi:10.1145/2077341.2077343.

Wang, Y., Tai, C., Sorkine, O., & Lee, T. (2008b). Optimized scale-and-stretch for image resizing. *ACM Transactions on Graphics, 27*(5), 1–8.

Ward, B., Kang, S., & Bennett, E. (2011). Depth director: A system for adding depth to movies. *IEEE Computer Graphics & Applications*, *31*(1), 36–48.

Zilly, F., Kluger, J., & Kauff, P. (2011). Production rules for stereo acquisition. *Proceedings of the IEEE*, *99*(4), 590–606.