Adaptive Estimation of Depth Map for Two-Dimensional to Three-Dimensional Stereoscopic Conversion

Fan GUO, Jin TANG, and Hui PENG

OPTICAL REVIEW Vol. 21, No. 1 (2014)

OPTICAL REVIEW Vol. 21, No. 1 (2014) 60–73

Adaptive Estimation of Depth Map for Two-Dimensional to Three-Dimensional Stereoscopic Conversion

Fan Guo, Jin Tang, and Hui Peng*

School of Information Science and Engineering, Central South University, Changsha, Hunan 410083, China

(Received May 30, 2013; Accepted September 26, 2013)

With increasing demands of three-dimensional (3D) contents, the conversion of many two-dimensional (2D) contents to 3D contents has become a focus in 3D image processing. The most important and difficult problem in 2D-to-3D conversion is the estimation of the depth map using only a single-view image. Therefore, a novel method for estimating a depth map for stereoscopic conversion is proposed in this paper. A simulated haze image generated by adding a haze veil on the input image is used to represent salient region segmentation. The 3D stereoscopic image is generated on the basis of the depth map estimated by the haze removal algorithm without any heuristic cues or user interaction. A comparative study and quantitative evaluation with some other state of the art conversion methods are carried out, which demonstrate that using the proposed approach, similar or better-quality results may be obtained.

Keywords: depth map, 2D-to-3D conversion, stereoscopic conversion, haze veil, saliency region, segmentation

1. Introduction

Rapid growth in the commercialization of three-dimensional (3D) displays has increased the demands for 3D media contents for supporting full-utility of 3D displays and has allowed us to experience more realistic and unique 3D effects. In addition to generating better visual experiences than conventional two-dimensional (2D) displays, emerging 3D displays have many applications in various fields, including broadcasting, film production, gaming, photography, camcorder design and production, and education. Owing to a lack of 3D media content, 2D-to-3D conversion could play a key role to convert existing 2D contents into 3D contents for growing 3D markets.

2D contents that require time-consuming manual editing of depth information have become a barrier to mass marketing, necessitating the development of an efficient 2D-to-3D conversion system. Thus, how to generate or estimate the depth map using only a single-view image has become the most important and difficult problem in 2D-to-3D conversion. Previous 2D-to-3D conversion methods are mainly divided into two classes: software-based methods and depth-cue-based methods. The software-based methods generate 3D content using stereoscopic conversion tools, such as DDD's TriDef and ArcSoft's Media Converter, to retrieve depth maps. However, although these software tools can automatically convert 2D content to 3D content, the stereoscopic visual effect produced by these tools is not obvious owing to the limited information they used for conversion.

A more feasible and effective method is the depth cue-based methods. This kind of method is based on the key observation that when observing the world, the human brain integrates various heuristic depth cues to generate depth perception. The major depth perceptions are binocular depth cues from two eyes and monocular depth cues from a single eye.[1] The disparity of the two images from eyes in the binocular visual system helps human eyes to converge and accommodate the object at the right distance. Monocular cues include focus/defocus, motion parallax, relative height/size, and texture gradient, providing various depth perceptions based on human experience. Therefore, humans can also perceive depth from a single-view image/video. The depth-cue-based method assigns depth values using image classification,[2] machine learning,[3] depth from focus/defocus,[4] depth from geometric perspective,[5] depth from texture gradient, depth from relative height,[6] and depth from multiscale local- and global-image features.

For example, Battiato et al.[2] generated depth maps using multiple cues from a single image. The depth is assigned on the basis of the properties of different regions. Hoiem et al.[3] generated maps using machine learning algorithms. The depths of scene and object are assigned according to the trained classes. However, the above methods are unreliable for cases missing during the train phase. Park and Kim[4] determined the distance from the focal plane using the blurriness of low depth-of-field in optical physics. However, depth estimation from focus/defocus cue estimates the depth range using edge blurriness are only feasible for small depth-of-field images and generate low-precision depth maps. Tsai et al.[5] obtained the depth gradient of a scene using the vanishing point. Jung et al.[6] assigned depth maps using edge information and prior depth knowledge. The computed image depth (CID) method[7] divides a single image into several sub-blocks and uses contrast and blurriness information to generate depth information for each block. Han and Hong[8] generated the depth map employing both vanishing points and super pixels as geometric and texture cues.

Although capable of exploiting various monotonic depth cues and appropriate for difficult cases, these methods are unreliable when the selected cues are weak in an input image. Cheng et al.[9] assigned the depth map on the basis of on a hypothesized depth gradient model. The method can produce impressive results. However, if the assumption of the global depth does not hold or large foreground objects

*E-mail address: huipeng@csu.edu.cn

exist, the method may fail. Yang et al.[10] generated a feasible perceptual depth map using the local depth hypothesis based on the structural information of the input image and salient regions. However, user interaction is required for this method.

Saxena et al.[11,12] proposed a discriminatively trained Markov random field (MRF) that incorporates multiscale local- and global-image features, and modeled the depths and the relationship between depths at different points in the image. However, the improved Gaussian and Laplacian MRFs are also proposed to incorporate both monocular cues and stereoscopic cues for depth map estimation. These algorithms can generate fairly accurate depth maps even on unstructured scenes, which is very important for the tasks in computer vision, such as autonomous obstacle avoidance and object detection.

2D-to-3D depth generation algorithms generally face two challenges. One is the depth uniformity inside the same object. A better segmentation of pixels implies a better outcome for the depth uniformity inside the object. An effective segmentation method should consider both color similarity and visual saliency. The other challenge involves retrieving an appropriate depth relationship among all objects. The estimated depth map should reflect the relative positions between scene objects and their neighboring regions. Generating a depth map from single 2D images is an ill-posed problem. Not all the depth cues can be retrieved from an image. To overcome these two challenges, in this work we present a novel algorithm that uses a haze veil to generate a pseudo depth map rather than retrieving the depth value directly from the depth cue. Firstly, the proposed algorithm produces a simulated haze image to represent salient region segmentation. Then the pseudo depth map is automatically generated in a single-view image using transmission information. Experimental results indicate that the proposed algorithm may generate promising stereoscopic results with slight side effects. The main contributions of this paper can be described as follows.

- A novel and automatic depth-map estimation method is proposed by adding a haze veil on the input image to represent salient region segmentation, and then fair stereoscopic images can be generated to provide the depth effect to the viewers.
- The proposed method is extended to video applications using the optical flow field, which greatly improves the temporal and spatial coherence of the depth map for the 2D-to-3D video conversion.
- By conducting appropriate qualitative and quantitative evaluation, the perception quality of the stereoscopic results is effectively measured.

The rest of this paper is organized as follows. Section 2 presents the objective of the proposed method. Section 3 describes an automatic 2D-to-3D conversion method using a simulated veil to generate the depth map. In Sect. 4, we extend the proposed algorithm to video application. To show the effectiveness of the proposed method, the experimental results and performance evaluation are given in Sect. 5, and conclusions are drawn in Sect. 6.

2. Objective

An image veil is proposed in this paper to segment the saliency region from a single input image. This veil is generated on the basis of the key observation that scene radiance is attenuated exponentially with depth, as indicated by the transmission map. If we can recover the transmission, then we can also recover depth information.[13,14] Thus, depth information can be measured using the transmission map. To date, numerous studies have been carried out to estimate the transmission map from a haze image. Therefore, if we may transform a haze-free image into a haze image for the purpose of 2D-to-3D conversion, then we may obtain depth information using various existing methods.

A simple and effective method for removing haze is based on the retinex theory.[15] On the basis of this theory, the input image I with haze is the product of object reflectance R, which can be regarded as a haze-free image, and scene illumination L, which can be regarded as a haze veil, that is:

$$I(x,y) = R(x,y) \cdot L(x,y), \tag{1}$$

where (x,y) is the position coordinate of a pixel. The main idea of the haze removal algorithm is to estimate the haze veil with the mean of the illumination component L that is obtained by convoluting the haze input image with a zero mean Gaussian smoothing function G. This process can be written as follows:

$$\hat{L}(x,y) = I(x,y) * G(x,y), \tag{2}$$

$$\tilde{L}(x,y) = \frac{1}{HW} \sum_{x=1}^{H} \sum_{y=1}^{W} \hat{L}(x,y), \tag{3}$$

where $\tilde{L}$ is the estimated haze veil, and H and W denote the height and weight of the image, respectively. The haze veil is subtracted from the original input image in the logarithmic domain to remove the haze effect from the input image, and then the exponential transformation is used to obtain the final haze-removed result $\tilde{R}$, as shown below.

$$\tilde{r}(x,y) = \ln I(x,y) - \ln \tilde{L}(x,y) = i(x,y) - l(x,y), \tag{4}$$

$$\tilde{R}(x,y) = \exp(\tilde{r}(x,y)). \tag{5}$$

Figure 1 shows a flowchart of the haze removal algorithm, and Fig. 2 illustrates an example of haze removal. From these figures, one can clearly see that the method uses the illumination component image obtained by the retinex algorithm to remove the veil layer from the input image.

Figure 3 illustrates the haze illusion by using two blocks to create the haze effect. Thus, we can deduce that the haze image [see Fig. 3(c)] is obtained by adding the haze-free input image [see Fig. 3(a)] to the haze veil [see Fig. 3(b)].

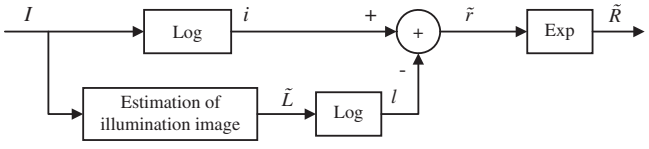


Fig. 1. Flowchart of the haze removal algorithm.

(a) (b) (c)

Fig. 2. (Color online) Illustration of the haze removal procedure. (a) Input image. (b) Estimated haze veil. (c) Haze removal result.

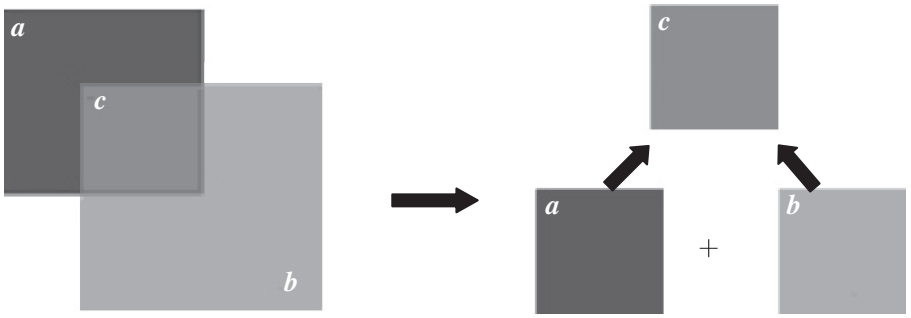


Fig. 3. Generation of the haze illusion.

Once the haze veil is derived from the input image, the haze effect is generated to compute the depth map of the input image.

3. Proposed Algorithm

3.1 Algorithm procedure

Specifically, the proposed veil algorithm has three steps to automatically convert a 2D image into a 3D one. The first step is to generate a simulated haze image by adding a haze veil on the haze-free input image, and the haze image is used to represent salient region segmentation. The second step is to compute the depth map by using the transmission map estimation in the haze removal algorithm, which includes initial depth map extraction, refined map estimation, and final depth map estimation. The goal of the algorithm is to generate a depth map without using any heuristic depth cues or any user interaction. Finally, the 3D stereoscopic image is generated on the basis of the estimated depth map. The overall procedure of this approach is shown in Fig. 4.

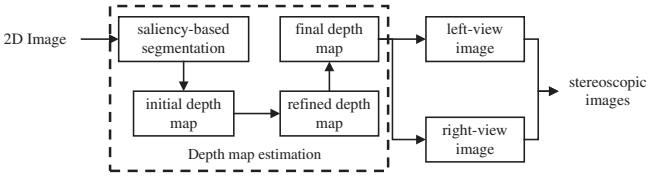


Fig. 4. Overall procedure for 2D-to-3D conversion.

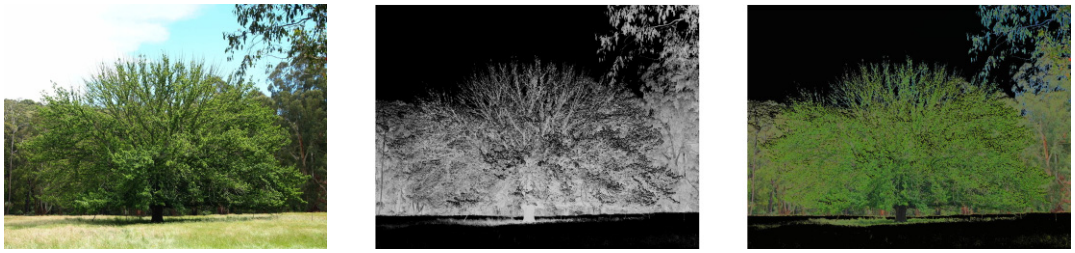
3.2 Pseudo depth map estimation

In general, the 2D-to-3D conversion from a single image has been assigned to the problem of how to generate depth-map information from 2D images. The depth map estimation is automatic and consists of the following four stages: haze image simulation, initial depth map extraction, refined map estimation, and final depth map estimation.

3.2.1 Haze image simulation

In this section, we propose a method for simulating a haze image by adding a haze veil on the haze-free input image. The theory behind the haze simulation process is that if the haze veil can be subtracted from the degraded image to remove haze (see Fig. 2), we can also simulate the haze image by adding the haze-free input image to the haze veil. In the haze-removal experiments, we find that the veil estimated through a mean calculation of illumination component can only handle a uniform-haze situation. If the haze is not uniform, color distortion of the haze-removal result often occurs. However, it is not always true that the haze is evenly distributed at each position since natural haze is dependent on unknown depth information. Thus, we present a new way to estimate a nonuniformly distributed haze veil in this paper.

According to the Koschmieder model,[16] the apparent luminance of the scene objects at different distances is different; therefore, different haze veils should be assigned according to their position. Therefore, we multiply the uniform veil $\tilde{L}$ by the intensity of the original image R and apply the color inversion operation to obtain a depth-like map. Considering that the intensity of an image reflects the amount of photons received by every position of an image, the smaller the distance between the scene points and the camera, the stronger the intensity will be; thus, the haze veil reflected by the depth-like map may be measured by its intensity. Therefore, we extract the intensity component of the depth-like map to produce the haze veil whose



(a) (b) (c)

Fig. 5. (Color online) Process of simulating haze image. (a) Original 2D image. (b) Haze veil. (c) Simulated haze image.

distribution determined the real fog density of the scene. Thus, the haze veil for the input image, $\tilde{L}'$ is estimated by

$$\tilde{L}'(x,y) = 255 - \omega_1 \times (R(x,y) \cdot \tilde{L}(x,y)), \tag{6}$$

where R is the input image without haze, $\tilde{L}$ is the mean of $\hat{L}$ obtained by Eq. (3), and ω_1 is an adjustment parameter set to 3 to generate a certain amount of haze in the input image. Then, we transform the image $\tilde{L}'$ from RGB to YCbCr color space, and extract the intensity component of the image, which stands for our final haze veil. Once the depth-like haze veil $\tilde{L}'$ is determined, the haze veil can be added on the real input haze-free image R to get the log-haze image $\tilde{\imath}$ after the conditions are set. The process is expressed as follows:

$$\tilde{\imath}(x,y) = \ln R(x,y) + \ln \tilde{L}'(x,y). \tag{7}$$

Finally, the simulated haze image I_{haze} can be obtained using exponential transformation, that is, $I_{\text{haze}} = \exp(\tilde{\imath}(x,y))$. The saliency region is segmented from nonsaliency regions (e.g., the sky and objects or surfaces that are too dark or too light) in the image I_{haze}, such that the haze image simulation is actually the image segmentation based on saliency. For example, Figs. 5(a) and 5(b) show the original 2D image and the estimated haze veil, respectively. The simulated haze image is shown in Fig. 5(c).

3.2.2 *Pseudo depth-map estimation*

Once the haze image I_{haze} is obtained, we can adopt the transmission estimation method that is widely used in haze removal to obtain depth information. For this purpose, the dark channel prior[13,14] and a guided filter[17] are used to estimate the depth map.

Specifically, we first estimate the atmospheric light A for the image I_{haze}. Most algorithms estimate A from the pixels with highest intensities, which is fast but not accurate. He et al.[13,14] integrated the atmospheric light estimation with the dark channel prior, which makes the estimation result more accurate. This method is also adopted in this paper.

The depth map is calculated on the basis of the image degradation model[16] and the dark channel prior proposed by He et al.[13,14] For the haze image, we first estimate the initial depth map $\tilde{m}(x,y)$. This process can be written as

$$\tilde{m}(x,y) = 1 - \omega_2 \min_{c\in\{R,G,B\}} \left(\min_{(x',y')\in\Omega(x,y)} \left(\frac{I^c_{\text{haze}}(x',y')}{A^c} \right) \right), \tag{8}$$

where I^c_{haze} is a color channel of I_{haze}, $\Omega(x,y)$ is a local patch centered at (x,y), and (x',y') is the pixel location that belongs to $\Omega(x,y)$. ω_2 is a constant parameter for adjusting the amount of haze for distant objects. The value of ω_2 is set to be 0.95 for all the results reported in this paper. More details on the initial depth map estimation are given in Appendix 1.

Note that there are obvious block effects and redundant details in the initial depth map. To overcome these deficiencies, we thus use a guided filter[17] and a bilateral filter to refine the initial depth map. The detailed estimation process of the final depth map is described in the following steps.

Step 1. For the initial depth map, we first compute the linear coefficients a_k and b_k for the guided filter:

$$a_k = \frac{\frac{1}{|\omega|}\sum_{(x,y)\in\omega_k} I_{\text{haze}}(x,y)\tilde{m}(x,y) - u_k\bar{m}_k}{\sigma_k^2 + \varepsilon}, \tag{9}$$

$$b_k = \bar{m}_k - a_k u_k$$

where I_{haze} is the guidance image and $\tilde{m}$ is the input image of the guided filter since the filter is a general linear translation-variant filtering process, which involves a guidance image and an input image.[13] In Eq. (9), ε is a regularization parameter keeping a_k from being too large. u_k and σ_k^2 are respectively the mean and variance of I_{haze} in a window ω_k centered at the pixel k. $|\omega|$ is the number of pixels in ω_k, and $\bar{m}_k = (1/|\omega|)\sum_{i\in\omega_k}\tilde{m}_i$ is the mean of $\tilde{m}$ in ω_k.

Step 2. Once the linear coefficients (a_k, b_k) are obtained, we can compute the filter output by

$$m'(x,y) = \bar{a}_k\tilde{m}(x,y) + \bar{b}_k, \tag{10}$$

where $\bar{a}_k = (1/|\omega|)\sum_{i\in\omega_k} a_i$ and $\bar{b}_k = (1/|\omega|)\sum_{i\in\omega_k} b_i$. $\tilde{m}$ is the initial depth map and the filter output m' is the refined depth map.

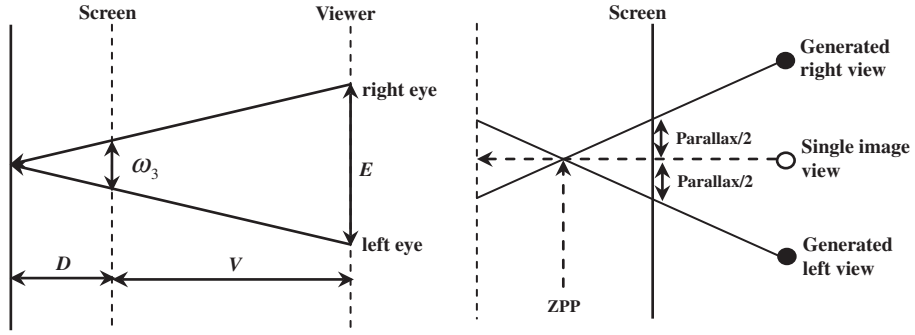
Step 3. A bilateral filter is used here to remove the redundant details for the refined depth map m' since the bilateral filter can smooth images while preserving edges. Thus, the redundant details of the refined depth map m' estimated by the algorithm presented above can be effectively removed. This process can be written as

$$\hat{m}(\mathbf{u}) = \frac{\sum_{\mathbf{p}\in N(\mathbf{u})} W_c(\|\mathbf{p}-\mathbf{u}\|)W_s(|m'(\mathbf{u})-m'(\mathbf{p})|)m'(\mathbf{p})}{\sum_{\mathbf{p}\in N(\mathbf{u})} W_c(\|\mathbf{p}-\mathbf{u}\|)W_s(|m'(\mathbf{u})-m'(\mathbf{p})|)}, \tag{11}$$

where $m'(\mathbf{u})$ is the refined depth map corresponding to the pixel $\mathbf{u} = (x,y)$ and $N(\mathbf{u})$ is the neighbor of $\mathbf{u}$. The spatial domain similarity function $W_c(x)$ is a Gaussian filter with the

(a) (b) (c)

Fig. 6. Process of estimating the depth map. (a) Initial depth map. (b) Refined depth map. (c) Final depth map.



(a) (b)

Fig. 7. Stereoscopic generation. (a) Max parallax computation. (b) Right-view and left-view generation.

standard deviation of σ_c: $W_c(x) = e^{-x^2/2\sigma_c^2}$, and the intensity similarity function $W_s(x)$ is a Gaussian filter with the standard deviation of σ_s, $W_s(x)$ can be defined as: $W_s(x) = e^{-x^2/2\sigma_s^2}$. In our experiments, the values of σ_c and σ_s are set to 3 and 0.4, respectively. Thus, we can obtain the final depth map $\hat{m}(x, y)$.

Figure 6 shows the corresponding initial depth map, refined depth map, and the final depth map for the original image in Fig. 5(a). From these figures, one can see that the final depth map [see Fig. 6(c)] generated using the proposed method reflects the relative positions between scene objects and their neighboring regions. Thus, the map is a pseudo depth map instead of a recovery of real depth information. Generally, the pseudo map is based on the visual attention of mapping the saliency regions from the position close to the viewer while mapping the nonsaliency regions from farther positions. Thus, images from the saliency region attract more visual attention and can be regarded as the final depth map for 2D-to-3D stereoscopic conversion.

3.3 3D image visualization using depth-map-based rendering

Once the depth map is obtained, the left-view and right-view images can be synthesized by the following steps. Firstly, we compute the parallax value Parallax(x, y) from each pixel (x, y) in the estimated depth map. The computation of the parallax value can be written as

$$\text{Parallax}(x, y) = \omega_3 \times \left(1 - \frac{\hat{m}(x, y)}{\text{ZPP}}\right), \tag{12}$$

where $\hat{m}(x, y)$ is the final depth map for the single image and ω_3 is the maximum parallax value. As can be seen in Fig. 7(a), we can obtain the value of ω_3 by a similar triangle principle. Specifically, V is the distance between the screen and the viewer, and the interocular distance E is about 6.35 cm. D is the maximum depth into the screen, and it is set to 10 cm. Thus, the computed ω_3 value is 0.578 cm. Next, we should express the value ω_3 in the form of pixels. In our experiment, a 17″ monitor (1280×1024 Resolutions) is used; therefore, 1 cm on the monitor is corresponds to 38 pixels. Thus, the maximum parallax value ω_3 is approximately 30 pixels for the image with a width of approximately to 1000 pixels. The zero parallax plane (ZPP) is set as the region with the depth value of Th, which is computed by $Th = \max(\hat{m}(x, y)) - 10$ to prevent separation and loss of artifacts.

Then, we consider the input image as the central view of the stereoscopic pair, as shown in Fig. 7(b). To produce the left- or right-view image, each pixel of the input image is shifted by the amount of Parallax$(x, y)/2$ in the left or right direction. The missing pixels at the image boundary will be filled to synthesize a right- or left-view image with the same size as the input original image. Finally, the anaglyph images can be generated by using these left- or right-view images. Viewers can experience the sense of depth with the help of anaglyph glasses (left: red; right: cyan) to view these images. For example, Figs. 8(a) and 8(b) are the left- and right-view images produced using the proposed approach for the input 2D image [Fig. 5(a)], respectively, and Fig. 8(c) shows our final 3D conversion result.

(a) (b) (c)

Fig. 8. (Color online) Process of 3D image visualization. (a) Left-view image. (b) Right-view images. (c) Stereoscopic image.

(a) (b) (c)

(d) (e) (f)

Fig. 9. (Color online) Process of generating stereoscopic images. (a) Input image. (b) Simulated haze image. (c) Estimated depth map. (d) and (e) Left-view and right-view images, respectively. (f) Stereoscopic conversion result.

Another example that illustrates the process of generating stereoscopic images is shown in Fig. 9. Here, virtual left- and right-eye views [Figs. 9(d) and 9(e), respectively] are rendered using the depth map [Fig. 9(c)] obtained from the proposed technique to evaluate performance. In most cases, the pseudo depth map is generated on the basis of the visual attention, as shown in Fig. 6(c). However, note that the estimated depth area corresponding to the person in Fig. 9(c) seems to be incorrect, as determined by saliency detection, but the approach may still provide good 3D perception, as shown in Fig. 9(f). This is because 2D-to-3D stereoscopic conversion does not require an accurate metric depth map, since human visual perception can generate correct results even when the depth map of an object is inverted. When the light gradient and the relative position between the salient objects and other parts of the scene are preserved, the human visual system may overwrite the depth perception with daily life experience. Hence, the light gradient and the relative position between the saliency region and nonsaliency region play an important role in depth perception. This could also explain why the side effects of the proposed algorithm are difficult to discover even when the saliency region or depth is inverted. Our observations on Figs. 9(c) and 9(f) confirm this conclusion.

4. Extension to Videos

For video applications, the primary interest is to enhance video, not single-image quality. For this purpose, we should ensure the temporal and spatial coherences of stereoscopic conversion results. Thus, we first extract the depth map frame-by-frame using the refined map estimation algorithm presented above, and then estimate the forward optical flow $\mathbf{u}_s^f$ and backward optical flow $\mathbf{u}_s^b$ between two neighboring frames to find the matched pixels. Finally, the flow fields are used to build an MRF model on the depth map to improve the temporal and spatial coherences of depth. Figure 10 shows the flow chart of the video processing framework. In the framework, the optical flow estimation algorithm of Sun et al.[18] is used to estimate the corresponding pixels between neighboring frames. With the forward and backward optical flows, we obtain the forward error map as $M_s^f = \|I(\mathbf{x}, s) - I(\mathbf{x} - \mathbf{u}_s^f, s-1)\|_2$, and the backward error map as $M_s^b = \|I(\mathbf{x}, s) - I(\mathbf{x} - \mathbf{u}_s^b, s+1)\|_2$. Here, $\|\cdot\|_2$ is the L_2-norm of a vector. The forward and backward error maps give a measure at each pixel of the accuracy of flow estimation from the previous and following frames. Figure 11 shows an example of forward and backward error maps.

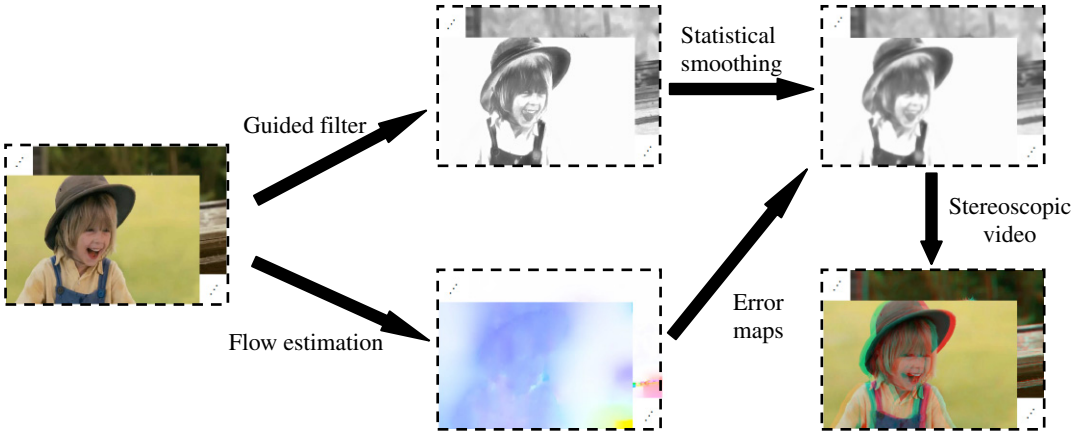


Fig. 10. (Color online) Video conversion flow chart.

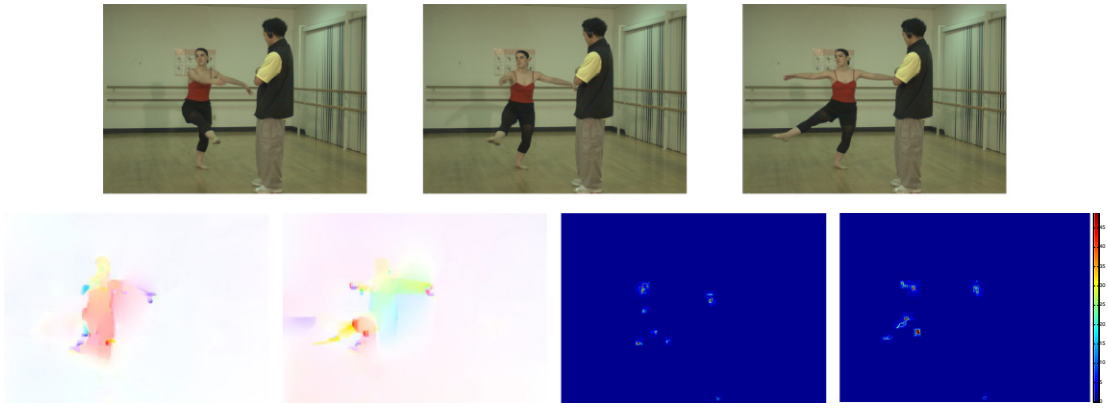


Fig. 11. (Color online) Forward and backward error maps. First row: frames 92, 93, and 94 of the original video. Second row: forward flow field, backward flow field, forward error map M_{93}^{f}, and backward error map M_{93}^{b}.

On the basis of the error maps, the MRF model is built to proceed with a statistical smoothing along the temporal dimension to improve the spatial and temporal coherences of the depth map $\hat{m}(\mathbf{x}, s)$ for each pixel $\mathbf{x}$ in each frame s. For a video with n frames, the MRF model is defined as

$$\begin{aligned} P(m) \propto & \prod_{\substack{s=1 \\ \mathbf{x}\in\mathbf{I}(\mathbf{x},s)}}^{n} \exp(-(m(\mathbf{x},s)-\hat{m}(\mathbf{x},s))^2/\sigma_p^2), \\ & \prod_{\forall \mathbf{y}\in\mathbf{\Omega}_x} \exp(-P_s(\mathbf{x},\mathbf{y},s)\cdot(m(\mathbf{x},s)-m(\mathbf{y},s))^2/\sigma_s^2), \\ & \prod_{\forall c\in\{f,b\}} \exp(-P_t(\mathbf{x},c,s)\cdot(m(\mathbf{x},s)-m(\mathbf{x}-\mathbf{u}_s^c,s'))^2/\sigma_t^2), \end{aligned} \tag{13}$$

where $\mathbf{\Omega}_x$ is the set of pixel $\mathbf{x}$'s four nearest-neighbors in the spatial domain, $\mathbf{I}(\mathbf{x}, s)$ represents each video frame s, and $\mathbf{u}_s^c$ is either forward flow $\mathbf{u}_s^f$ or backward flow $\mathbf{u}_s^b$. When $c = f$, $s' = s - 1$, and when $c = b$, $s' = s + 1$. $P_s(\mathbf{x}, \mathbf{y}, s)$ is the spatial prior $P_s(\mathbf{x}, \mathbf{y}, s) = 1/(\hat{m}(\mathbf{x}, s) - \hat{m}(\mathbf{y}, s))^2$, and $P_t(\mathbf{x}, c, s)$ is the temporal prior that can be written as $P_t(\mathbf{x}, c, s) = 1/M_s^c(\mathbf{x}), \forall c \in \{f, b\}$. For each pixel $\mathbf{x}$ in frame s, $\mathbf{x} - \mathbf{u}_s^c$ is the corresponding pixel in frame s'. The first term in Eq. (13) defines the expectation of the final depth map, and the calculation is carried out for the corresponding pixel in each frame s. The second term describes the spatial prior in each frame and the last one describes the temporal prior of adjacent frames. We maximize the probability by solving the linear system resulting from $d \log P/dm = 0$ and take this optimum to be the final depth map m for each video frame. Thus, the 3D video sequences can be obtained from each 2D frame with improved video coherence. More details are given in Appendix 2.

5. Experimental Results

5.1 Qualitative comparison

5.1.1 Image test

Figure 12 shows some experimental results for single images, including data on seven sets of the original images, depth maps, and red-cyan images. The method generates fair stereoscopic images, which offer a good depth effect to viewers, as shown in Figs. 12(a)–12(g).

Figure 13 shows the comparison of our results with those obtained using the Laplacian model of Saxena et al.[11,12] The data used in the experiment are available on the Internet database.[19] Note that the results obtained with our 2D-to-3D

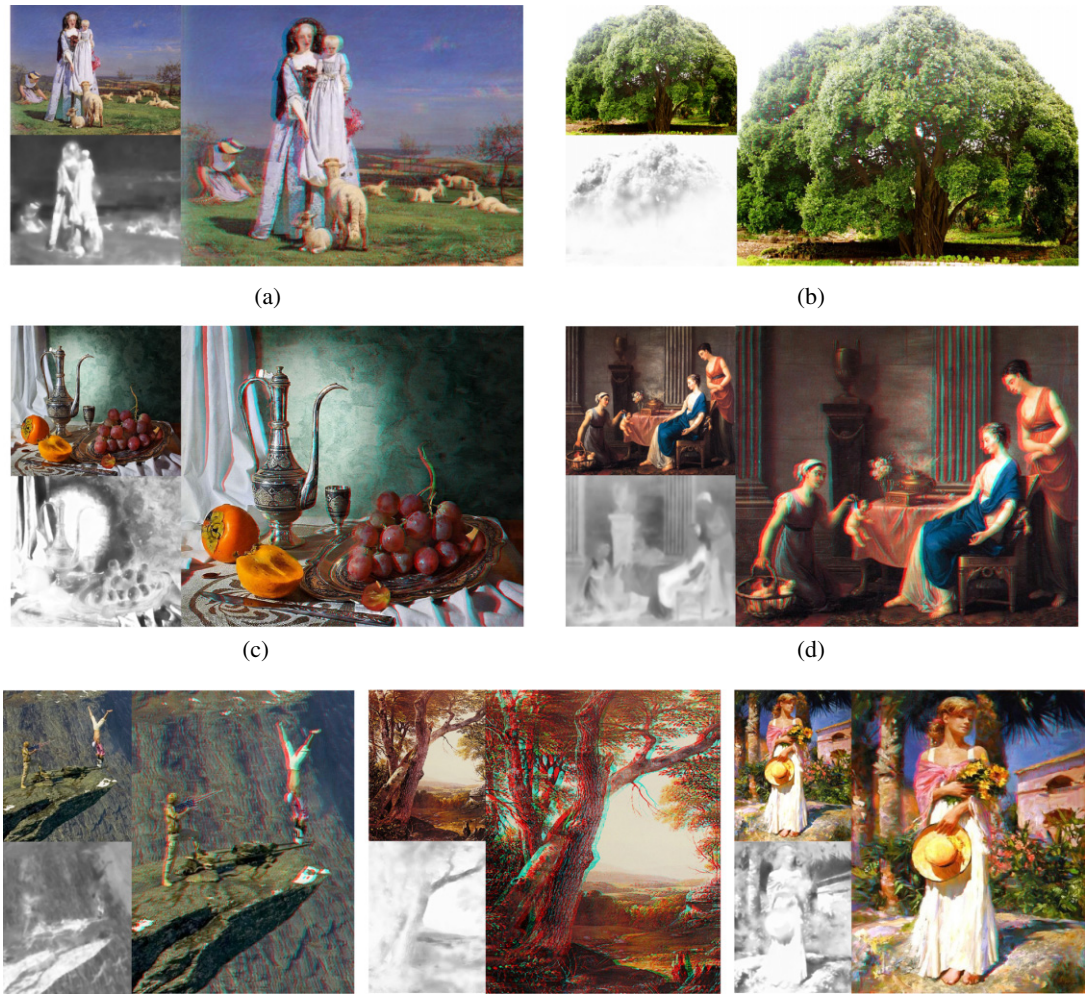


Fig. 12. (Color online) Examples of 2D-to-3D stereoscopic conversion results for a set of images.

conversion seem visually close to the results obtained using the Laplacian model of Saxena et al., with less separation and loss of artifacts than the results generated by the ground truth depth map, owing to the guided and bilateral filters, which respectively capture the depth boundary and remove the redundant details at the same time. The reason for the slightly worse performance of the ground truth depth map lies in the boundary mismatch between the original images and the corresponding depth map for 2D-to-3D stereoscopic conversion. Owing to the mismatch problem, synthesized views generate separation and loss of the foreground objects, which sometimes result in visual artifacts, as shown in Fig. 13. In contrast, the proposed algorithm achieves better results than the Laplacian model of Saxena et al. as illustrated in the experiments. However, the main difference between the two methods is that the algorithm of Saxena et al. is a supervised learning approach that incorporates both local and global image features to predict the depth map with a training set of monocular images, whereas the proposed algorithm in this paper is an unsupervised learning approach that generates the depth map by simulating haze as the global image feature without using a training set.

5.1.2 *Video test*

For video application, a *breakdancing* sequence from Microsoft Research was used to perform the qualitative evaluation. The public dataset provided the depth map of each frame that was computed using the stereo technique [see Fig. 14(b)].[20] Figure 14(c) shows the stereoscopic conversion results generated by the provided depth map, and Fig. 14(d) shows the conversion results obtained using our estimated depth map. From the figure, one can see that similar 3D stereoscopic effects can be obtained by using the two methods. However, our proposed method uses only a single image to generate depth, whereas the depth map in the previous work[20] is generated using a stereo algorithm with at least two captured images to interpret the depth values.

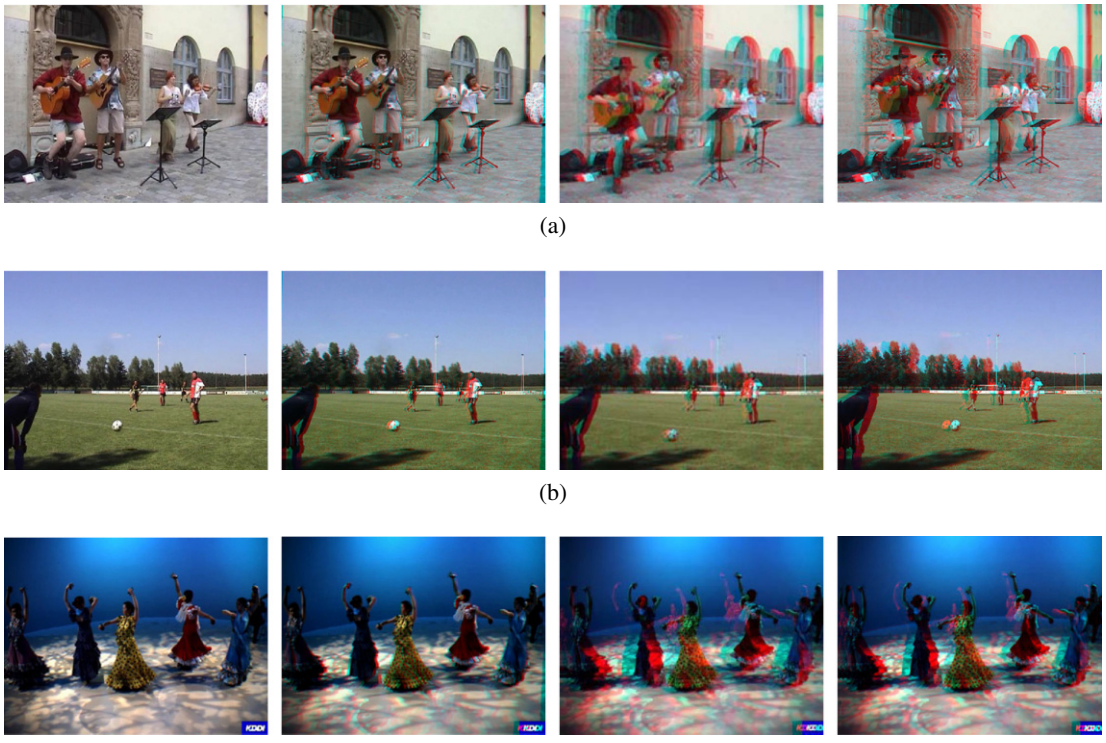
We also evaluate the visual quality of the proposed algorithm by comparison with other 2D-to-3D conversion approaches: the commercial software of ArcSoft's Media Converter 7 and the method of Cheng et al.[9] The test video database consists of three sequences, *barden*, *fussball*, and *flamingo*. Figure 15 shows the comparison of generated red-cyan images for the three test sequences. From the figure,

Fig. 13. (Color online) Comparison results for a set of various scenes. Original images and corresponding ground truth depth maps from Internet database[19] (row 1 and row 2), stereoscopic results obtained using the ground truth depth map (row 3), estimated depth map generated using Laplacian model of Saxena et al.[11,12] and corresponding stereoscopic results (row 4 and row 5), and our estimated depth map and corresponding stereoscopic results (row 6 and row 7).

one can see that the 3D effects produced by Media Converter 7 are not obvious compared with the results generated using the other two algorithms, since the simple and easy-to-use media converter mainly utilizes the color cue to extract depth information. The algorithm of Cheng et al.[9] can produce vivid and realistic visual effects. However, a hypothesized depth gradient model is required for the method. When the assumption of the global depth does not

(a) (b) (c) (d)

Fig. 14. (Color online) Examples of 2D-to-3D stereoscopic conversion results for a video clip. (a) Original video frames. (b) Estimated depth map in previous work.[20] (c) Stereoscopic results obtained using the provided depth map. (d) Stereoscopic results obtained using our estimated depth map.



(a)

(b)

(c)

Fig. 15. (Color online) Comparison of red-cyan images for three test sequences. First column: original 2D image. Second Column: results obtained using Arcsoft's Media Converter 7. Third column: results obtained by the method of Cheng et al.[9] Fourth column: results obtained by the proposed method. (a) *Barden* sequence, (b) *Fussball* sequence, (c) *Flamingo* sequence.

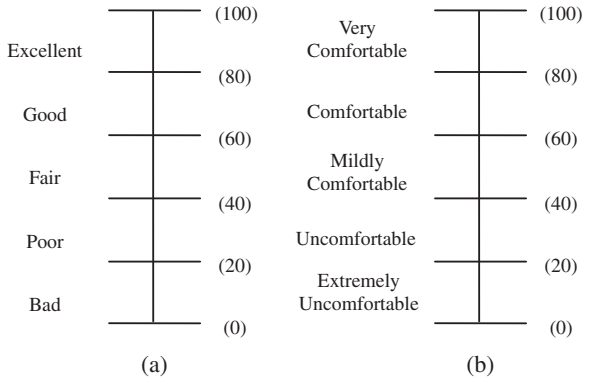


Fig. 16. Rating scales used for evaluation of (a) depth quality and (b) visual comfort.

hold or large foreground objects exist, the depth gradient hypothesis is invalid. Note that the results obtained using our algorithm seem visually close to the results obtained using the algorithm of Cheng et al. without using any hypothesis or user interaction, owing to the haze image simulation. Figure 15 shows some examples of red-cyan stereoscopic results generated by different methods.

5.2 *Quantitative evaluation*

To quantitatively assess and rate the different 2D-To-3D conversion methods, the seven images in Fig. 12, the four images in Fig. 13, and the three video sequences in Fig. 15 were used to perform the quantitative evaluation. Both depth quality and visual comfort were evaluated using a single — stimulus presentation method that is a slightly modified version of that described in previous works.[9,21] Depth quality measures the sense of depth experienced by the viewer, and visual comfort refers to the subjective sensation of comfort that accompanies the physiological change.[22] Thus, depth quality and visual comfort can both be measured by asking the viewer to report his/her level of perceived depth quality and visual comfort. The synthesized red-cyan images were viewed using anaglyph glasses. The subjective evaluation was performed by 10 individuals with normal or correct-to-normal visual acuity and stereoacuity. The participants watched the stereoscopic images or videos in a random order and were asked to rate each image or video on the basis of two factors, depth quality and visual comfort. The overall depth quality was assessed using a five-segment scale, as shown in Fig. 16(a), and visual comfort was assessed using that shown in Fig. 16(b).

Figures 17(a) and 17(b) show the values of the two factors acquired using Media Converter 7 and our proposed algorithm for the test images shown in Fig. 12. From Fig. 17, one can deduce that in terms of depth quality, viewers can experience a better sense of depth using the proposed algorithm compared with the results obtained using ArcSoft's software.

Figures 18(a) and 18(b) show the comparisons of our conversion results with the ground truth and the results of Saxena et al. for the test images shown in Fig. 13. From

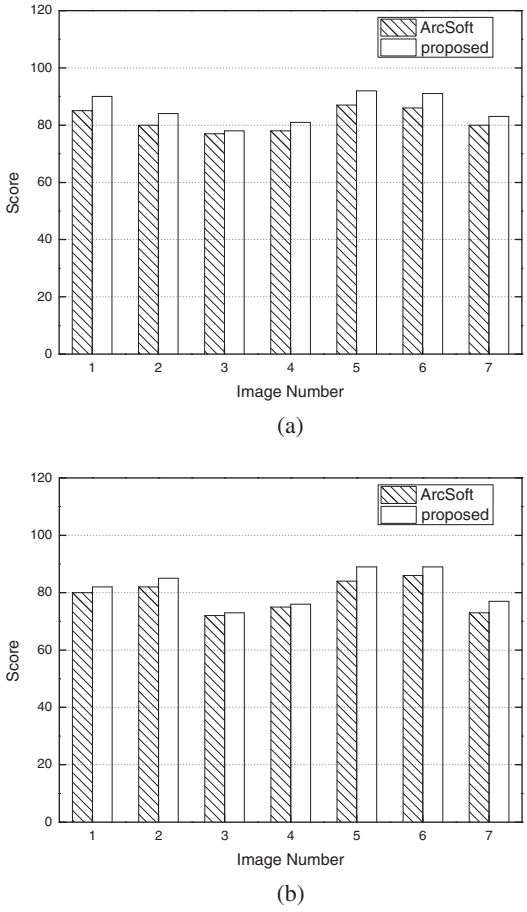
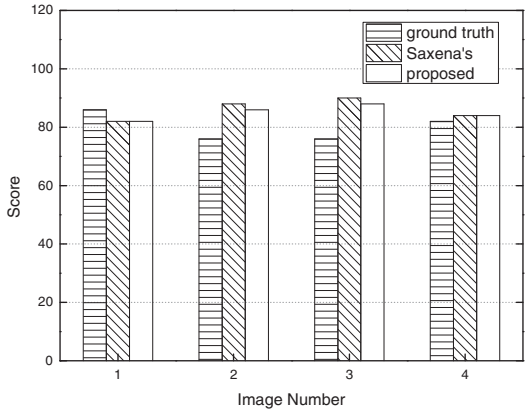


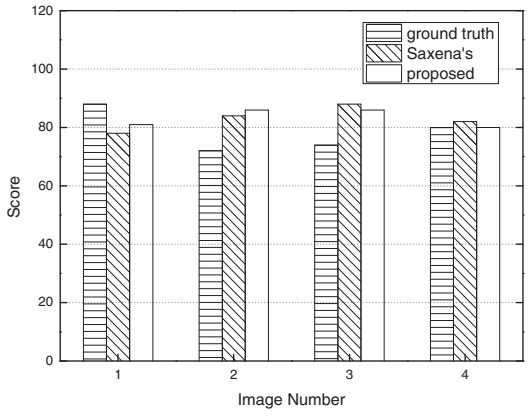
Fig. 17. Quantitative evaluation results for test images shown in Fig. 12. (a) Depth quality, (b) visual comfort.

Fig. 18, one can see that, compared with the results obtained using the ground truth depth map, the algorithm of Saxena et al. and the proposed algorithm have fewer side effects and yield slightly better quality in general. For example, in column 2 (Fig. 13), the tree trunk in the stereoscopic results of the ground truth depth map has obvious visual artifacts. The same artifacts can also be observed in column 3 in Fig. 13. Therefore, for depth quality and visual comfort, the result obtained using the ground truth depth map scores worse in this case.

One can also clearly see that the depth map estimated by the proposed method looks completely different from that estimated by Saxena et al. However, the scores of depth quality and visual comfort are similar between these methods. According to Tam et al.,[22] the factors that affect visual comfort can be divided into five categories: (a) accommodation-vergence conflict, (b) binocular mismatches, (c) depth inconsistencies, (d) parallax distribution, and (e) cognitive inconsistencies. The testing images used for comparison were viewed on the same displays and by the same observers from the same distance that equally optimized picture quality. Thus, there is no accommodation-vergence conflict for both methods to cause a difference in depth quality and visual comfort. Furthermore, the left- and right-view images of both methods are produced by

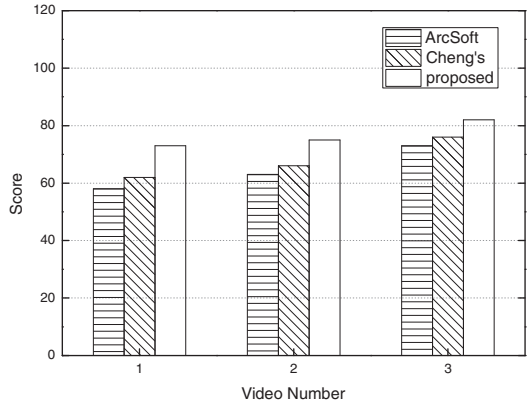


(a)

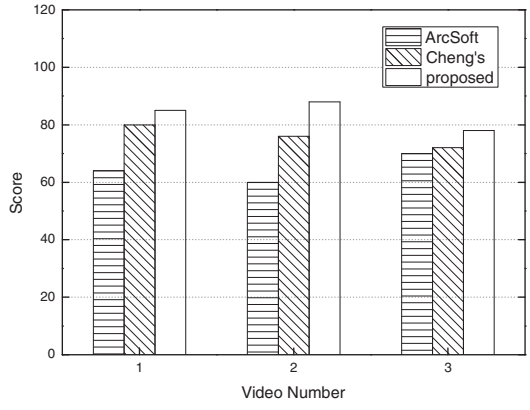


(b)

Fig. 18. Quantitative evaluation results for test videos shown in Fig. 13. (a) Depth quality, (b) visual comfort.



(a)



(b)

Fig. 19. Quantitative evaluation results for test videos shown in Fig. 15. (a) Depth quality, (b) visual comfort.

leftward or rightward shifting of each pixel of the input image to a certain parallax value [see Eq. (12)]. Since the 2D input image and its corresponding depth map of both methods are all matched properly, there is no binocular mismatch problem to cause visual discomfort. Depth inconsistency is also one of the factors that affect visual comfort. The depth maps estimated by both methods inevitably contain erroneous information, which might cause depth inconsistencies. Nevertheless, it is not known to what extent such inconsistencies would affect visual comfort.[22] For the fourth factor, parallax distribution of Saxena et al. measured using the depth map seems closer to the ideal distribution in which the bottom portion of the image appeared closer and the top portion of the image appeared farther away. However, from the experiment, we find that perceptual and cognitive consistencies are the most important factors in 2D-to-3D stereoscopic conversion. Although the depth map estimated by the proposed method only provides a new way to simulate the virtual left- or right-view image and it does not recover the real depth information, the human visual system may overwrite the depth perception and make the depth cues consistent with our daily life experience. In other words, our eyes are tricked by our brain. To the best of our knowledge, there has been no formal investigation that has attempted to explain why the side effects of these methods are hard to observe even when the depth is inverted. Since for both methods there is no conflicting information between the disparity information contained in the stereoscopic image and the depth cues that are normally experienced in the real world, the proposed method have scores in depth quality and visual comfort similar to those of Saxena et al. Notably, depending on the image, any algorithm can obtain the highest score in depth quality or visual comfort.

For the video sequence, we compare our results with other two conversion methods ArcSoft's software and the method of Cheng et al.[9] for the three test videos shown in Fig. 15. Evaluation results demonstrate that the proposed algorithm produces a similar or better 3D visual effects compared with Cheng's method for the input video, as shown in Fig. 19. For instance, in *fussball* and *flamingo*, one can see that the proposed algorithm has the best scores in depth quality and visual comfort. This confirms our observations in Figs. 15(b) and 15(c). Furthermore, when the images vary in terms of the lighting source, the proposed algorithm still works well. This is because the illumination on the surface is well preserved in the proposed method, so humans have correct depth perception with daily life

experience. Thus, we can rank the three methods for the test video in decreasing order with respect to visual comfort: the proposed method, the method of Cheng et al. and ArcSoft's software.

6. Conclusions

A novel and automatic method was proposed to generate a pseudo depth map in a single-view image using an estimated haze veil. A haze image was simulated by adding a haze veil on the input image to represent salient region segmentation and estimate the pseudo depth map by the transmission estimation method in the haze removal algorithm. Using the depth map, left- and right-view images were synthesized, and finally the stereoscopic images were generated to provide a sense of depth to the viewers with the help of anaglyph glasses. The whole process of the proposed algorithm could be performed automatically without any heuristic cues or user interaction. Our future work includes improving the computational efficiency of the proposed algorithm using GPU implementation or parallel computation.

Acknowledgements

This work was supported by the International Science and Technology Cooperation Program (2011DFA10440), the National Natural Science Foundation of China (71271215, 70921001, and 91220301), the Postdoctoral Science Foundation of Central South University (No. 126648), and the New Teacher Fund for School of Information Science and Engineering, Central South University (No. 2012170301).

Appendix 1: Initial Depth Map Estimation

The model of the effect of fog is established by Koschmieder using the following relationship:[16]

$$I_{\text{haze}}(x,y) = J(x,y)e^{-\beta d(x,y)} + A(1 - e^{-\beta d(x,y)}), \qquad (A{\cdot}1)$$

where $I_{\text{haze}}(x,y)$ is the apparent luminance at pixel (x,y), $d(x,y)$ is the distance between the corresponding objects with intrinsic luminance $J(x,y)$, A is the atmospheric light, and β denotes the extinction coefficient of the atmosphere. Let $\tilde{m}(x,y) = e^{-\beta d(x,y)}$. $\tilde{m}(x,y)$ is called the transmission map[13,22] and can be regarded as a pseudo depth map, since it indicates the relationship between the scene radiance and the scene depth. Thus, one can rewrite Eq. (A·1) as a simpler image formation model in the RGB color space,

$$I^c_{\text{haze}}(x,y) = J^c(x,y)\tilde{m}(x,y) + A(1 - \tilde{m}(x,y)) \quad \text{for } c \in \{R,G,B\}. \qquad (A{\cdot}2)$$

Taking the minimization operation in the local patch $\Omega(x,y)$ in Eq. (A·2), the model can be rewritten as

$$\min_{c\in\{R,G,B\}}\left\{\min_{(x',y')\in\Omega(x,y)}\left(\frac{I^c_{\text{haze}}(x',y')}{A}\right)\right\} \qquad (A{\cdot}3)$$
$$= \tilde{m}(x,y)\min_{c\in\{R,G,B\}}\left\{\min_{(x',y')\in\Omega(x,y)}\left(\frac{J^c(x',y')}{A}\right) + (1 - \tilde{m}(x,y))\right\}.$$

Ideally, the minimum intensity in a patch of the ideal haze-free image J^c should be zero according to the dark channel prior.[13,14] A constant parameter ω_2 is also introduced to make the image more natural. Therefore, the pseudo depth map can be computed by

$$\tilde{m}(x,y) = 1 - \omega_2 \min_{c\in\{R,G,B\}}\left(\min_{(x',y')\in\Omega(x,y)}\left(\frac{I^c_{\text{haze}}(x',y')}{A^c}\right)\right). \qquad (A{\cdot}4)$$

Appendix 2: Derivation for MBF Optimization

The framework for estimating the optimized depth map of an input video is as follows. First, the pseudo depth map $\hat{m}(x,y)$ is extracted frame by frame. Then, the forward optical flow $\mathbf{u}^f_s$, the backward optical flow $\mathbf{u}^b_s$, the forward error map M^f_s, and the backward error map M^b_s between two neighboring frames are estimated to find the matched pixels. Finally, the flow fields are used to build an MRF model on the pseudo depth map to obtain the final depth map for the input video, which greatly improves the spatial and temporal coherences of the depth map. Thus, optimizing the depth map for an input video sequence can be achieved by computing $d\log P/dm = 0$. More formally, from Eq. (13), we have

$$\begin{aligned}\log P(m) \propto & \sum_{\substack{s=1\\ \mathbf{x}\in I(\mathbf{x},s)}}^{n}(-(m(\mathbf{x},s) - \hat{m}(\mathbf{x},s))^2/\sigma_p^2) \\ & + \sum_{\forall \mathbf{y}\in\mathbf{\Omega}_\mathbf{x}}(-P_s(\mathbf{x},\mathbf{y},s)\cdot(m(\mathbf{x},s) - m(\mathbf{y},s))^2/\sigma_s^2) \\ & + \sum_{\forall c\in\{f,b\}}(-P_t(\mathbf{x},c,s)\cdot(m(\mathbf{x},s) - m(\mathbf{x} - u^c_s, s'))^2/\sigma_t^2),\end{aligned} \qquad (B{\cdot}1)$$

where $\mathbf{\Omega}_x$ is the set of pixel $\mathbf{x}$'s four nearest-neighbors in spatial domain, $\mathbf{I}(\mathbf{x},s)$ represents each video frame s, and $\mathbf{u}^c_s$ is either forward flow $\mathbf{u}^f_s$ or backward flow $\mathbf{u}^b_s$. σ_p^2, σ_s^2, and σ_t^2 are constant parameters. When $c = f$, $s' = s - 1$, and when $c = b$, $s' = s + 1$. $P_s(\mathbf{x},\mathbf{y},s)$ is the spatial prior, $P_s(\mathbf{x},\mathbf{y},s) = 1/(\hat{m}(\mathbf{x},s) - \hat{m}(\mathbf{y},s))^2$, and $P_t(\mathbf{x},c,s)$ is the temporal prior that can be written as $P_t(\mathbf{x},c,s) = 1/M^c_s(\mathbf{x})$, $\forall c \in \{f,b\}$. For each pixel $\mathbf{x}$ in frame s, $\mathbf{x} - \mathbf{u}^c_s$ is the corresponding pixel in frame s'. For obtaining the final depth map, and for the following constraint to be valid

$$\begin{aligned}& m(\mathbf{x},s) \approx \hat{m}(\mathbf{x},s), \quad m(\mathbf{y},s) \approx \hat{m}(\mathbf{y},s), \\ & m(\mathbf{x} - \mathbf{u}^c_s, s') \approx \hat{m}(\mathbf{x} - \mathbf{u}^c_s, s'),\end{aligned} \qquad (B{\cdot}2)$$

we have

$$\begin{aligned}d\log P/dm \propto & \sum_{\substack{s=1\\ \mathbf{x}\in I(\mathbf{x},s)}}^{n}\left(-\frac{2}{\sigma_p^2}(m(\mathbf{x},s) - \hat{m}(\mathbf{x},s))\right) \\ & + \sum_{\forall \mathbf{y}\in\mathbf{\Omega}_\mathbf{x}}\left(-\frac{2P_s(\mathbf{x},\mathbf{y},s)}{\sigma_s^2}\cdot(m(\mathbf{x},s) - \hat{m}(\mathbf{y},s))\right) \\ & + \sum_{\forall c\in\{f,b\}}\left(-\frac{2P_t(\mathbf{x},c,s)}{\sigma_t^2}\cdot(m(\mathbf{x},s) - \hat{m}(\mathbf{x} - u^c_s, s'))\right) = 0.\end{aligned} \qquad (B{\cdot}3)$$

Thus, the final depth map for the input video sequence can be inferred as

$$m(\mathbf{x},s)=\left(\sum_{\substack{s=1\\ \mathbf{x}\in I(\mathbf{x},s)}}^{n}\frac{2}{\sigma_p^2}\cdot\hat{m}(\mathbf{x},s)+\sum_{\forall\mathbf{y}\in\Omega_{\mathbf{x}}}\left(\frac{2P_s(\mathbf{x},\mathbf{y},s)}{\sigma_s^2}\cdot\hat{m}(\mathbf{y},s)\right)\right.$$
$$\left.+\sum_{\forall c\in\{f,b\}}\left(\frac{2P_t(\mathbf{x},c,s)}{\sigma_t^2}\cdot\hat{m}(\mathbf{x}-\mathbf{u}_s^c,s')\right)\right)\Bigg/\left(\sum_{\substack{s=1\\ \mathbf{x}\in I(\mathbf{x},s)}}^{n}\frac{2}{\sigma_p^2}+\sum_{\forall\mathbf{y}\in\Omega_{\mathbf{x}}}\frac{2P_s(\mathbf{x},\mathbf{y},s)}{\sigma_s^2}+\sum_{\forall c\in\{f,b\}}\frac{2P_t(\mathbf{x},c,s)}{\sigma_t^2}\right). \quad \text{(B·4)}$$

Once the output m is obtained, we iteratively take m into Eq. (B·4) as $\hat{m}$ until the condition $|m_{num}(\mathbf{x},s)-m_{num-1}(\mathbf{x},s)|<\varepsilon'$ is fulfilled, where num is the number of iterations ($num\geq 2$) and ε' is a matrix with the same small value. We thus have $m(\mathbf{x},s)=m_{num}(\mathbf{x},s)$ defined as the final depth map for the input video sequence. In the experiments, the iteration processes will converge after 3 or 4 times in most cases.

References

1) W. J. Tam and L. Zhang: Proc. ICME, 2006, p. 1869.
2) S. Battiato, S. Curti, E. Scordato, M. Tortora, and M. La Cascia: Three-Dimensional Image Capture and Applications V1 **5302** (2004) 95.
3) D. Hoiem, A. A. Efros, and M. Hebert: Proc. ACM SIGGRAPH, 2005, p. 577.
4) J. Park and C. Kim: Proc. SPIE VCIP, 2006, p. 60771O-1.
5) Y.-M. Tsai, Y.-L. Chang, and L.-G. Chen: Proc. Intl. Symp. Intelligent Signal Proc. and Comm. Syst., 2006, p. 586.
6) Y. J. Jung, A. Baik, J. Kim, and D. Park: Proc. SPIE - Int. Soc. Opt. Eng., 2009, p. 72371U-1.
7) H. Murata, Y. Mori, S. Yamashita, A. Maenaka, S. Okada, K. Oyamada, and S. Kishimoto: Proc. SID Digest of Technical Papers 32.2, 1998, p. 919.
8) K. Han and K. Hong: Proc. ICCE, 2011, p. 651.
9) C.-C. Cheng, C.-T. Li, and L.-G. Chen: IEEE Trans. Consum. Electron. **56** (2010) 1739.
10) N.-E. Yang, J. W. Lee, and R.-H. Park: Proc. ICCE, 2012, p. 311.
11) A. Saxena, S. H. Chung, and A. Y. Ng: Proc. NIPS, 2005, p. 1161.
12) A. Saxena, S. H. Chung, and A. Y. Ng: Int. J. Comput. Vis. **76** (2008) 53.
13) K. M. He, J. Sun, and X. O. Tang: Proc. CVPR, 2009, p. 1956.
14) K. He, J. Sun, and X. Tang: IEEE Trans. Pattern Anal. Mach. Intell. **33** (2011) 2341.
15) E. H. Land: Proc. Natl. Acad. Sci. U.S.A. **83** (1986) 3078.
16) N. Hautiére, J.-P. Tarel, J. Lavenant, and D. Aubert: Mach. Vision Appl. **17** (2006) 8.
17) K. M. He, J. Sun, and X. O. Tang: Proc. ECCV, 2010, p. 1.
18) D. Sun, S. Roth, and M. J. Black: Proc. CVPR, 2010, p. 1.
19) Make3D Laser+Image data (Dataset-2): http://make3d.cs.cornell.edu/data.html.
20) C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski: ACM Trans. Graph. **23** (2004) 600.
21) ITU-R Recommendation BT.500-10, (2000).
22) W. J. Tam, F. Speranza, S. Yano, K. Shimono, and H. Ono: IEEE Trans. Broadcast **57** (2011) 335.
23) M. van Rossum and Th. Nieuwenhuizen: Rev. Mod. Phys. **71** (1999) 313.