Interactive Stereoscopic Video Conversion

Zhebin Zhang, Chen Zhou, Yizhou Wang, and Wen Gao, *Fellow, IEEE*

***Abstract*—This paper presents a system of converting conventional monocular videos to stereoscopic ones. In the system, an input monocular video is firstly segmented into shots so as to reduce operations on similar frames. An automatic depth estimation method is proposed to compute the depth maps of the video frames utilizing three monocular depth cues—depth-from-defocus, aerial perspective, and motion. Foreground/background objects can be interactively segmented on selected key frames and their depth values can be adjusted by users. Such results are propagated from key frames to nonkey frames within each video shot. Equipped with a depth-to-disparity conversion module, the system synthesizes the counterpart (either left or right) view for stereoscopic display by warping the original frames according to their disparity maps. The quality of converted videos is evaluated by human mean opinion scores, and experiment results demonstrate that the proposed conversion method achieves encouraging performance.**

***Index Terms*—2-D-to-3-D (stereoscopic) video conversion, depth estimation, stereoscopic video quality evaluation.**

I. Introduction

DUE TO THE amazing development of the 3-D television (3DTV) industry (e.g., broadcasting of many 3-D channels), the number of available stereoscopic videos is largely inadequate to satisfy the great demand of the market even with the new-make of such videos using stereo cameras. Converting conventional 2-D videos into stereo ones is definitely a complimentary solution. However, automatic 2-D-to-3-D video conversion remains a challenging problem. This is mainly because the core problem—depth from monocular view—has not been solved. Although there exist techniques to estimate depth from 2-D image sequences, such as structure from motion (SfM) [36], it requires special conditions such as constrained camera motion, foreground stillness, and rigidity assumptions. These constraints make a large proportion of real videos falls beyond its regime. In addition, in order to obtain a good estimate, these algorithms always require high quality intermediate results such as good feature matching and motion estimation. Whereas, in real videos, such requirement may be too demanding due to all kinds of variations, degradations, and occlusions.

Manuscript received August 21, 2012; revised December 6, 2012, February 5, 2013 and March 20, 2013; accepted April 17, 2013. Date of publication June 17, 2013; date of current version September 28, 2013. This work has been supported by the grant of National Basic Research Program of China (973 Program) 2009CB320904, NSFC 61121002, NSFC 61272027, CPSF 2013M530482, NSFC 61231010, NSFC 61210005 and NSFC 91120004. This paper was recommended by Associate Editor P. Callet.

The authors are with National Engineering Laboratory for Video Technology and Key Laboratory of Machine Perception (MoE), School of EECS, Peking University, Beijing 100871, China (e-mail: zbzhang@pku.edu.cn; chzhou@pku.edu.cn; yizhou.wang@pku.edu.cn; wgao@pku.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2013.2269023

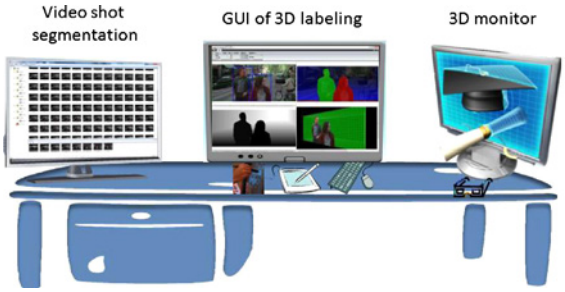

Fig. 1. Interfaces of the stereoscopic conversion system.

A. System Overview

In this paper, an interactive system is proposed to convert monocular videos into stereoscopic ones (Fig. 1). We introduce human in the loop only to rescue the deficiency of the state-of-the-art automatic algorithms. The proposed system includes the following main parts:

1) Before the conversion, the system segments an input monocular video into shots so that the depth and the labeled information such as foreground object contours can be propagated efficiently and reliably within each shot.
2) A novel multicue depth estimation method is proposed. It integrates a few robust monocular depth perception cues adopted by human beings, such as depth from defocus, depth from aerial perspective, and depth from motion. Considering each cue has its own algorithmic flaw in depth estimation (e.g., motion estimation can be unreliable in textureless regions), we compute the confidence of each cue at every superpixel and set the superpixels of high confidence as depth anchors. Then the depth values from different cues are aligned to the same depth range, and the depth values of the anchor superpixels are propagated to the rest through a Markov random field (MRF). In addition, we propose to use occlusion boundaries (OB) derived from the motion cue to resolve the defocus ambiguity problem [28]. These cues largely improve the depth estimation performance by mitigating the constraints of the SfM method and its extensions (e.g., [22] and [36]). Using the proposed method, the range of the estimated depth becomes much wider than SfM-based methods.
3) To improve the stereoscopic visual quality, we adopt interactive methods to segment foreground objects, and estimate their 3-D shape and depth positions in particular.
4) We propose to integrate depth information in a tracking algorithm so as to improve the foreground object tracking accuracy.

5) A depth-to-disparity conversion model based on supervised learning is proposed, which predicts object disparity according to its motion, screen location, and the background motion. It automatically generates inward-/on-/outward-screen stereo visual effects learned from stereoscopic movies.

In summary, the proposed interactive 2-D-to-3-D conversion system is easy to operate. The conversion method is robust and efficient, and it adapts to a wide range of depth estimation from monocular videos.

The rest of the paper is organized as follows: related work is introduced in Section II. In Section III, we describe the workflow and architecture of the system, followed by expatiations of important functions and methods in Section IV. In Section V, evaluation and experimental results are provided. Finally, Section VI concludes the paper.

II. Related Work

In the literature, the methods of 2-D-to-3-D conversion can be roughly categorized into two classes, the automatic conversion and interactive conversion. Automatic conversion methods (see [17], [18], [26], [30], [41], and [42]) exploit motion cues to predict the depth/disparity of pixels. Commercial software, such as DDD-TriDef 3-D player and Samsung's 3DTV, leverage both motion and scene geometric priors to generate stereoscopic views from monocular videos. However, motion cues such as optical flows can be unreliable to extract in real-image sequences, and motion parallax can be ambiguous in estimating relative depth between objects in a complex dynamic system (i.e., multiple objects moving with different velocities at different depths). In addition, assumptions of specific scene geometry can be too strong to be true, e.g., the bottom region of an image is closer to view point compared to the upper part. In conclusion, fully automatic conversion methods usually are incompetent to estimate scene depth accurately given the state-of-the-art computer vision algorithms.

The other category of conversion methods exploit user interactions [10], [22], [43] . For example, in [10] and [22], at key frames, user scribbles are used to initialize depth values and depth layers of objects in a scene, and then the depth information is automatically propagated to nonkey-frames. IMAX developed a sophisticated commercial interactive conversion system [16], which requires intensive manual work and can generate impressive stereoscopic visual effects. Compared to these methods, the proposed model integrates more monocular cues in depth estimation, so that it adapts to a more variety of real videos and generates a wider estimation of depth range.

Single view depth estimation is a crucial step in 2-D-to-3-D video conversion and it is yet an active challenging research topic in computer vision. Researchers took advantage of various cues for this task, such as photometry cues (e.g., [13], [28], [38]), geometry cues (e.g., [7], [9]), motion cues (see [17], [18], [26], and [30]), and appearance cues (e.g., [10], [32]). However, to our best knowledge, a robust and integrated framework that ingeniously combines different depth cues are expected to be proposed. In the following, we review these works even though some of them has not been applied to stereo video conversion yet.

Photometry cues: Objects in an image usually are not all in focus [28], especially when they are captured by professional camera lens, e.g., prime lens. Valencia *et al.* [38] used wavelet analysis and edge defocus estimation to obtain the relative depth of the pixels in an image. However, such methods can only be applied to the case when the focused object is frontmost . If the focused object is in the middle, the other objects in front of and behind it are all blurred in the captured image. The degree of blur only indicates the relative distance to the focused object but not to the camera [as shown in Fig. 8(a)]. This phenomenon is called ambiguity of defocus in depth estimation. In this paper, we use OB to resolve the ambiguity (see Section IV-B3).

Besides, scene atmospheric light also facilitates depth perception. Atmospheric radiance images of outdoor scenes are usually degraded by the turbid medium in the atmosphere. Irradiance received by a camera from a scene point is attenuated along the line of sight. He *et al.* [13] proposed a dark channel prior to remove haze and also provided estimated scene depth maps as a by-product.

Geometry cues: Parallel edge lines converging at infinity due to perspective projection provide us a convenient geometry formulation to reconstruct the relative distance between objects, e.g., [7], [9], [12].

Motion cues: Under the condition of constrained camera motion and assuming that scenes are static, there are two ways to estimate disparity maps, using 1) SfM (e.g., [36]), and 2) motion parallax (e.g., [17]). However, in real scenarios such as movies, the constrained camera motion condition and static scene assumption are often violated, which leads to the failure of applying the two methods.

Appearance cues: By using appearance features from superpixels (e.g., color, texture), Hoiem *et al.* [14] casted the depth estimation from single images to a multilabel classification problem. Saxena *et al.* [32] employed MRF to model unstructured scenes and directly learned the relationship between 3-D structures and texture and color features. Both methods adopted supervised learning, which requires training data to learn the model; this makes their models heavily dependent on the training data sets.

III. Proposed System

A. Workflow

The system starts with video-audio splitting followed by video decoding. Then it converts the video to stereo. The converted video is compressed by a stereo video encoder—the stereo high profile in the H.264/AVC reference software of JM-18.0 [1]. Finally the system merges the coded video with the original audio into a demanded format. The key part of the system is the 2-D-to-3-D conversion module, which consists of the following parts as shown in Fig. 2.

1) The multicue depth estimation module predicts depth value of each pixel using multicue depth inference and generates scene depth maps.

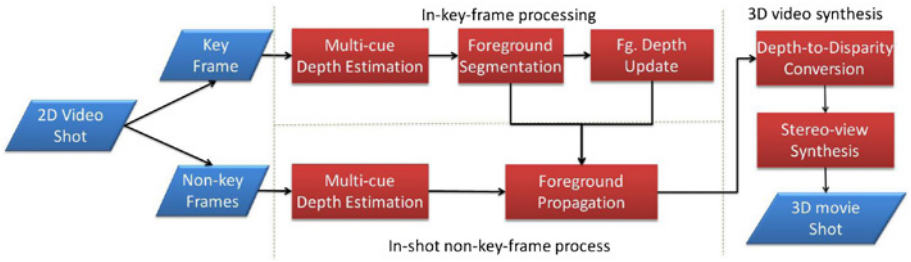

Fig. 2. Flowchart of the 2-D-to-3-D conversion module.

2) The foreground/background segmentation module segments foreground objects and background regions on key frames using user scribbles.
3) The foreground depth update module updates the depth values of foreground pixels according to the relative positions of the objects w.r.t. the background.
4) The foreground propagation module tracks foreground objects in nonkey-frames and propagates the depth of the foreground pixels.
5) The depth-to-disparity conversion module adopts a trained model to automatically predict pixel disparities based on their motion and depth.
6) The stereo-view synthesis module generates another view of the video so as to render the stereoscopic visual effects.

The system is designed to reduce user interaction as much as possible. Hence, before a video is sent to the 2-D-to-3-D conversion module, the video is segmented into shots so that the labeled information can be reliably propagated within a shot as shown in Fig. 4.

B. Interfaces and Interactions

The proposed system interacts with users through the following three interfaces (shown in Fig. 1).

The shot segmentation interface (shown in Fig. 4) Users can use it to select certain shots to process.

The 3-D labeling interface [shown in Fig. 3(a)] It consists of four windows. In Window 1, the user scribbles on foreground objects and their vicinity regions of background in a video frame, so that foreground objects are segmented semiautomatically. Window 2 displays the estimated depth map of a frame. In Window 3, both foreground objects and background are displayed in a 3-D grid, so that users can clearly see their relative depth relationship in 3-D space. In addition, a virtual screen (the green plane) is provided to demonstrate their relative position w.r.t. to the display (i.e., whether an object is inward, on or outward the screen). The depth value of each pixel as well as the virtual screen are manually adjustable. Hence, users can easily correct depth estimation errors and render desired stereoscopic effect. Window 4 shows a color map, which demonstrates the inward-/on-/outward-screen distribution of the objects. Green areas indicate that the regions are on the screen, read ones are outward, and blue ones are inward.

The stereoscopic monitor Users can examine the rendered stereoscopic effects of converted frames. If the results are not satisfactory, users can refine them by adjusting the labels through the 3-D labeling interface in Window 3.

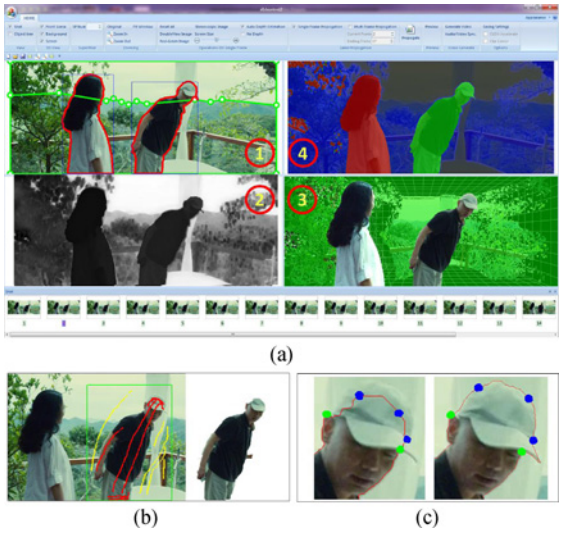

Fig. 3. Interactive segmentation of foreground objects.

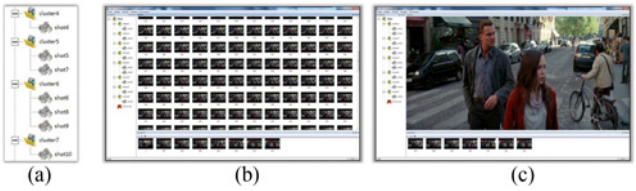

Fig. 4. Video shot segmentation interface.

IV. 2-D-to-3-D Conversion Methods

As mentioned above, the proposed system first segments an input video into shots, then converts each shot to stereo by following the steps shown in Fig. 2. In the rest of the section, we introduce the details.

A. Shot Segmentation

Video shots are defined as a set of meaningful and manageable segments, which share the same background setting [21]. Consequently, information can be easily propagated within a shot. Our video segmentation algorithm consists of the following steps: 1) feature extraction; 2) dissimilarity computation between frames; and 3) shot boundary detection.

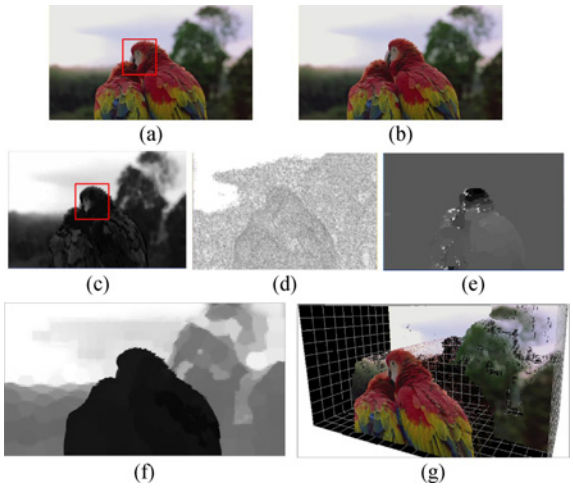

Fig. 5. Depth estimation by multicue fusion. (a) and (b) Two consecutive frames from a source video. (c)–(e) Depth maps by three depth cues. The darker the color of a pixel, the closer it is to the camera. (c) Algorithmic error: it predicts white/gray regions farther away than the object's actual depth. (f) Fused depth map. (g) Predicted pixels rendered in 3-D grid.

Feature extraction Both the color histograms of the original frame[35] and of the tiny image [37] are used as features. (A tiny image is an image with a resolution of 24×12 down sampled from its original resolution). Specifically, a color histogram in RGB space with 16 bins for each channel is computed from a frame.

Dissimilarity computation In [23], Liu *et al.* defined a dissimilarity metric of two images as a combination of L_1 norm distance between their color histograms and mutual information. In this paper, we also consider the contribution of tiny images.

Let $C_i(k)$ and $T_i(k)$ denote the color histogram and tiny image pixel value for the ith frame respectively, where k is one of the K_* possible values ($K_C = 48$ for color histogram and $K_T = 24 \times 12$ for tiny image in our case). Then the L_1 norm distances between two color histograms and two tiny images are defined as $\Gamma_C(i, j) = \sum_{k=1}^{K_C} |C_i(k) - C_j(k)|$ and $\Gamma_T(i, j) = \sum_{k=1}^{K_T} |T_i(k) - T_j(k)|$. The mutual information between the ith and the jth frame is computed based on color, $MI(C_i, C_j) = H(C_i) + H(C_j) - H(C_i, C_j)$, where $H(C_i)$ is the entropy of colors in the ith frame and $H(C_i, C_j)$ denotes the joint entropy of colors in two frames.

Finally, the dissimilarity between two frames is

$$\Gamma(i, j) = w_C \times \frac{\Gamma_C(i, j)}{MI(C_i, C_j)} + w_T \times \Gamma_T(i, j) \tag{1}$$

where $w_C = w_T = 0.5$ in our implementation.

Shot boundary detection To decide shot boundaries, an adaptive segmentation threshold at frame t is proposed as

$$T(t) = \eta \times \frac{\sum_{i=2}^{t} \Gamma(i, i-1)}{t-1} \tag{2}$$

where i indexes the ith frame of a shot. The threshold is an averaged dissimilarly up to frame t multiplying with a factor. η is introduced to avoid under-segmentation ($\eta = 4.0$ in our implementation). When $\Gamma(t, t-1) > T(t)$, a shot boundary is marked at t.

B. Depth Estimation by Multicue Fusion

In this paper, we deliberately select three depth perception cues to estimate the initial depth maps of video frames, namely the motion cue, defocus cue, and aerial perspective cue (as shown in Fig. 5). Each of them governs a different range of depth estimation. SfM method is able to accurately estimate scene depth at a near distance; defocus cue is good to predict a mid-range depth; and aerial perspective cue can give a reasonable estimate of scene depth at a far distance. Although each cue alone cannot reliably recover the depth of a scene due to its algorithmic flaws (e.g., motion cue is not applicable to textureless regions, a white or gray colored object in front will be labeled as a distant object if using the aerial perspective cue, and the depth ambiguity in defocus cue), the combination of the three usually is able to provide a robust depth estimation by compensating the weakness of each other.

We denote the depth maps of a frame I predicted by the three cues as $\alpha_m(I)$, $\alpha_d(I)$, $\alpha_a(I)$ ($\alpha_m(I)$ is from the motion cue, $\alpha_d(I)$ is from the defocus cue and $\alpha_a(I)$ is from the aerial perspective cue), and let (x, y) be the coordinates of a pixel p on I. (The coordinate of upper left corner of an image is $(0, 0)$.) In the following paragraphs, we first introduce the depth estimation algorithm of each cue, and how to overcome the algorithmic flaws by selecting high confidence estimation on superpixels as depth anchors. Then we describe a method that fuses the depth estimation from the three cues into a final depth map.

1) *Depth from Aerial Perspective Cue:* As described in [13], the irradiance attenuates along the sight in a scene, so the depth of a pixel p is taken as

$$\alpha_a(p) = -\epsilon \ln t(p) \tag{3}$$

where ϵ is the scattering coefficient of the atmosphere. $t(p)$ is the medium transmission, which is estimated using the dark-channel prior proposed in [13]

$$t(p) = 1 - \omega \min_c (\min_{p' \in \Omega} \frac{I^c(p')}{A^c}) \tag{4}$$

where ω is a constant parameter $0 < \omega < 1$, p' is a pixel in a local patch centered at p, $I^c(p)$ is the intensity value in the color channel c (in RGB color space), and A^c is the atmospheric light intensity (please refer to [13] for the details of estimating A^c). The depth of a superpixel, $\alpha_a(s)$, is just the mean depth of its inside pixels. (In the paper, the SLIC method [2] is used to obtain the superpixels of an image.)

Depth Anchors However, this algorithm has inherent flaws—it assumes that brighter regions are closer to the camera than white/gray regions. Thus the algorithm predicts correct depth for white/gray regions when they are at a far distance, but always over-estimates their depth when they are close [e.g., the white face of the parrot in Fig. 5 (a) and (c)]. Considering such defect, we introduce a confidence measure for the estimates of the aerial perspective cue according to

the color and location of pixels. For a pixel p, the confidence value is defined as

$$v_a(p) = \begin{cases} 1, & y < h_1 \\ \max(1 - \frac{y}{h_2}, g(p)), & h_1 \le y < h_2 \\ g(p), & \text{else} \end{cases} \quad (5)$$

where y is the vertical coordinate of a pixel. In this paper, we set $h_1 = \frac{1}{4}h$ and $h_2 = \frac{1}{3}h$ (h is the image height). $g(p)$ is defined as

$$g(p) = \frac{v(p)}{\max_{q \in I} v(q)} \quad (6)$$

where $v(p) = \frac{1}{255}\max(|c_r - c_g|, |c_b - c_g|, |c_r - c_b|)$, and c_r, c_g, c_b are the intensity values of the RGB channels, respectively. The denominator of (6) is the maximal value of $v(p)$ in image I, and it normalizes the confidence value to the range of [0, 1]. If an object is white or gray (i.e., the RGB values are similar.), $g(p)$ values are small; If an object is bright (i.e., there is a big difference between the RGB channels.), $g(p)$ values are large.

We assume that pixels in the lower part of an image ($y \ge h_2$) are closer to the camera. However, if there is a white or gray object in this region (of small $g(p)$), the algorithm will predict a large depth value [13], thus we assign the estimation with a low confidence value ($v_a(p) = g(p)$). For bright objects in this region (of large $g(p)$), the algorithm will correctly predict its distance, so we assign a large confidence ($v_a(p) = g(p)$) to the prediction. The upper part of an image ($h < h_1$) tends to be the sky, which is far away, and the dark channel prior [13] usually predicts the pixels' distances correctly. Thus we assign their confidence to $v_a(p) = 1$. For a pixel in the middle part of an image ($h_1 \le y < h_2$), it is very easy to confuse whether a white/gray pixel belongs to a foreground object or background sky [e.g., the parrot white face in Fig. 5(a) and (c)]. We assign the confidence values of these pixels to the larger ones between $1 - \frac{y}{h_2}$ and $g(p)$, which balances their color and image locations. This spatial layout prior can be restrictive when converting unconventional scenes. In our system, we make it an option for this prior in the user interface, so that user can choose to use it during a conversion.

To compute the confidence value of a superpixel s, we average all the pixels' confidence values in s

$$v_a(s) = \frac{1}{|s|}\sum_{p \in s} v_a(p). \quad (7)$$

For the superpixels with higher confidence, we set them as depth anchors and use them to fuse the depth values from other cues (in Section IV-B4). In this paper, the depth anchors from the aerial perspective cue are selected if $v_a(s_i) > 0.5$, which is an empirical threshold in the experiments.

An example of an estimated depth map from the areal perspective cue and its the depth anchors are shown in Fig. 6(a) and (l), respectively.

2) *Pseudodepth from Motion:* Here we make a very simple assumption that an object with smaller motion is further away from the viewing point

$$\alpha_m(p) = 1 - \frac{m(p)}{\max_{p \in I} m(p)} \quad (8)$$

where $m(p)$ is the motion magnitude at pixel p. Although the assumption is simple, it is generally true especially for background, since the foreground objects are dealt separately in the system at a later stage (see Section IV-C).

The depth of a superpixel, $\alpha_m(s)$, is just the mean depth of its inside pixels.

Depth Anchors: It is known that the estimation of motion parallax or depth from stereo matching depends on feature correspondences. Hence, the estimation is more trustworthy in textured regions than in textureless regions. Here, we set the depth estimation confidence of a region using the motion cue according to the richness of texture in the region

$$v_m(s) = \frac{1}{|s|}\frac{\#_{kp}(s)}{\max_{t \in I} \#_{kp}(t)} \quad (9)$$

where $|s|$ is the size or the number of pixels in superpixel s, and $\#_{kp}(s)$ returns the number of key points in superpixel s (kp stands for "key points." Here we use Harris corner point [11] as the key point.) Examples of an estimated depth map and depth anchors from the motion cue are shown in Fig. 6(e) and (m).

3) *Depth from Defocus:* In movies, cameramen often take advantage of focus/defocus skills to enhance visual effect, e.g., closeups. Thus depth from defocus can be applied to many video shots.

Image defocus can be considered as a heat diffusion process [19]

$$\begin{cases} \frac{\partial u(p,t)}{\partial t} = c\,\frac{\partial^2 u(p,t)}{\partial p^2} \\ u(p,0) = f(p) \end{cases} \quad (10)$$

which assumes that at the initial state, an image $f(p)$ is all-focused. Then, at $t = \tau$ it is diffused to blur. The blurred image is the solution to the heat diffusion equation (10), $I(p) = u(p, \tau)$. In (10), c is called the diffusion factor, which relates to the blurring parameter [8],

$$\sigma^2 = 2tc \quad (11)$$

where σ is the parameter of the Gaussian blurring kernel and a larger value of σ indicates a farther distance to the camera. Hence, the diffusion time t provides a clue to depth estimation (assuming it is an anisotropic diffusion with constant c).

In this system, we estimate depth from defocus by simulating a reversed heat diffusion process introduced in [27]

$$\begin{cases} \frac{\partial u(p,t)}{\partial t} = -\beta(p)\; c\;\frac{\partial^2 u(p,t)}{\partial p^2} \\ u(p,0) = I(p). \end{cases} \quad (12)$$

i.e., we take the blurred image $I(p)$ as the initial state of the process, and reverse the diffusion process till all the pixels are sharp. $\beta(p)$ is an indicator for stopping the diffusion process on a pixel p

$$\beta(p) = \begin{cases} 1, & \text{if } |\nabla u(p) - \overline{\nabla u}| < \theta \\ 0, & \text{else} \end{cases} \quad (13)$$

i.e. when the intensity gradient of a pixel, $\nabla u(p)$, is similar to the average gradient of its 8-connected neighbor pixels, $\overline{\nabla u}$, the reverse diffusion stops. ($\theta \in [0.2, 0.4]$ in our implementation.)

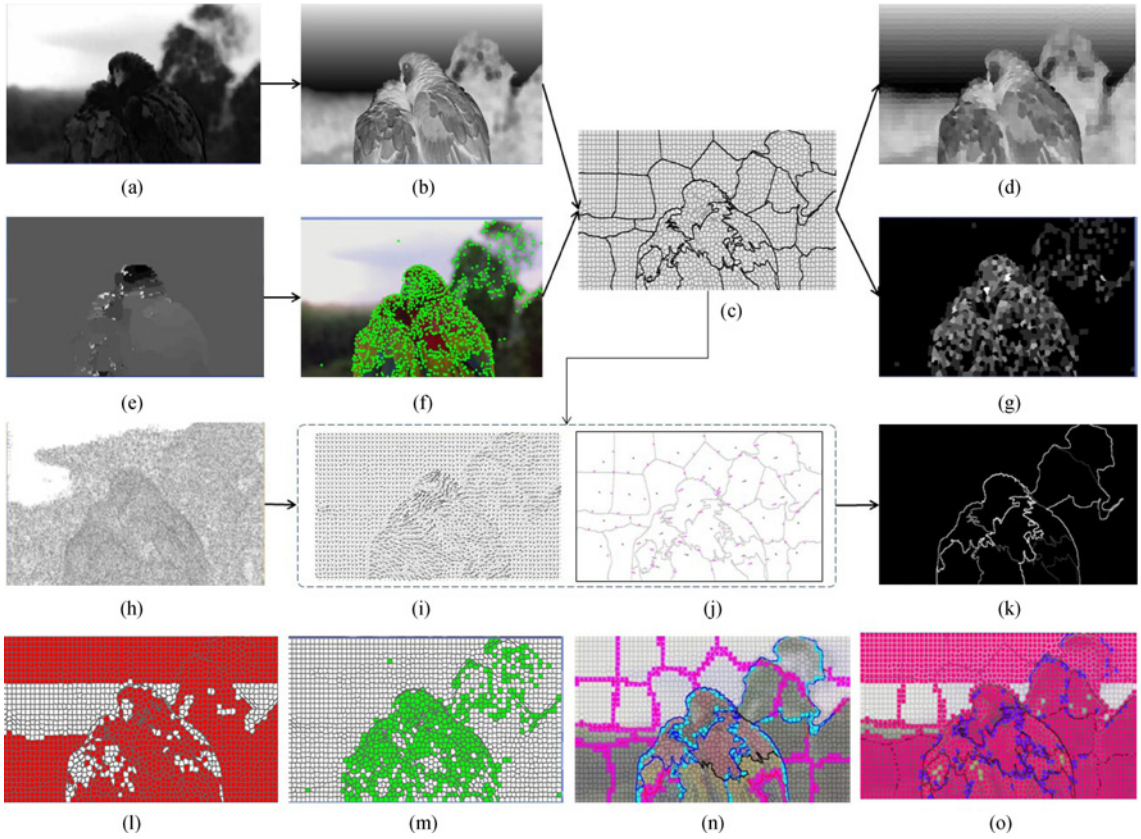

Fig. 6. Generating depth anchors (using Fig. 5(a) as an example). The depth maps from the three cues, aerial perspective (a) motion parallax (e) and defocus (h), are first computed. (The darker the color of a pixel, the closer it is to the camera.) For the aerial perspective cue, based on the pixel-wise confidence map (b) and the over-segmentation (c) superpixel confidence map can be computed. For the motion cue, Harris corners are detected (f) and the confidence map of motion cue (g) is computed by counting the key points number in each superpixel. For the defocus cue, the motion fields of two layers of over-segmentation [(i) for superpixels and (j) for regions] are used to compute the occlusion boundary confidence map (k). Given the corresponding thresholds of the aerial perspective cue and the motion cue, the anchor points with high confidence are selected (l) and (m), respectively. For the defocus anchors (n), cyan anchors are the superpixels attached to the inner side of OB (blue boundaries), and pink anchors are the ones attached to the region boundaries (the red boundaries). (o) All the anchors from the three cues—among them the purple ones are the common anchors of the three cues.

Then the relative depth for each pixel is computed as

$$\tilde{\alpha}_d(p) = \int_0^{t(p)} c\, dt' = c \cdot t(p). \tag{14}$$

$t(p)$ is the stopping time at pixel p. As it is relative depth, we assume $c = 1$ without loss of generality. (We shall align the depth range from different cues in Section IV-B4.) Fig. 6(h) shows a depth map obtained using the reversed diffusion on a blurred image.

As shown in Fig. 6(h), the pixel-wise depth estimation from defocus is noisy. We use the soft-max value of the pixels within a superpixel as the relative depth of the superpixel

$$\tilde{\alpha}_d(s_i) = \frac{\sum_{p \in s_i} \tilde{\alpha}_d(p) e^{\tilde{\alpha}_d(p)/\varsigma}}{\sum_{p \in s_i} e^{\tilde{\alpha}_d(p)/\varsigma}} \tag{15}$$

where ς is a constant and set to 0.8 in this paper. Untill now the defocus ambiguity (Section II) has not been resolved in the current relative depth map.

Depth Anchors: For the defocus cue, according to the estimation algorithm, we consider that the estimated depth on edges is more accurate. There are two types of edges in an image, texture edges (TE) on an object, and object OB. For TE, the depth values can be propagated on both sides of the edges. Hence, the depth anchors are selected on both sides of the edges, which is trivial [e.g., the pink superpixels in Fig. 6(n)]. Whereas for OB, their depths is determined by the object, so the anchors are on the object side [e.g., the cyan superpixels in Fig. 6(n)].

In the following paragraphs, we first introduce how to detect OB and assign boundaries to the correct objects. Then, we discuss how to resolve the defocus ambiguity.

1) *Occlusion boundary detection and boundary assignment.* As shown in Fig. 7, we first compute the motion field from two close frames (3–5 frames apart), and get mean motions for large image regions and region boundaries [Fig. 7(c)]. The image regions can be obtained by any image segmentation method. We assume that a sufficient condition of being an occlusion boundary is that the motions across the boundary are different. Hence, we define the confidence value of being an occlusion boundary by the dissimilarity of the motions across the edge

$$v_o(b_i) = 1 - \frac{\mathbf{m}_{i_1} \cdot \mathbf{m}_{i_2}}{|\mathbf{m}_{i_1}||\mathbf{m}_{i_2}|} \tag{16}$$

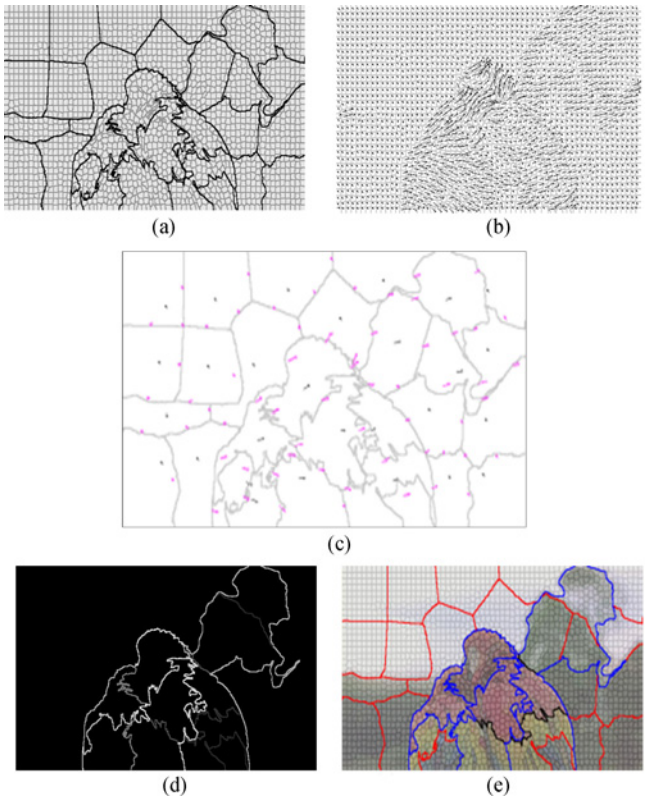

Fig. 7. Motion-based occlusion boundary detection. The SLIC method[2] is adopted to obtain superpixels. (a) With different segmentation parameters, a two-layer segmentation is obtained at a fine scale and a coarse scale. We call the smaller ones as "superpixels" and the larger ones as "regions." (b) Superpixel motion vectors are computed from the optical flow of the pixels inside each superpixel. (c) Region motions (black arrows) are obtained from the superpixel motions inside each region, and the boundary motions (purple arrows) are computed from the pixel motions on the boundaries. (d) Brighter color indicates a higher probability of a boundary being an occlusion boundary. (e) Detected OB (blue) and region boundaries (red). A black color indicates that the type of a boundary is undecided.

where b_i is a region boundary, $\mathbf{m}_{i_1}$ and $\mathbf{m}_{i_2}$ are mean motion vectors of the two regions. One example of occlusion boundary confidence map of a frame is shown in Fig. 7(d). The boundaries with high confidence value (e.g., $v_o(b_i) > 0.6$) are selected as OB, e.g., the blue boundaries in Fig. 7(e). And the boundaries with small confidence (e.g., $v_o(b_i) < 0.4$) highlighted with red are considered as TE; they are inside of an object. Otherwise, the type of a boundary is undetermined boundary marked with black in Fig. 7(e). (In this paper, the threshold 0.6 for OB and the threshold 0.4 for TE are chosen empirically.)

We assume that the boundary shares the same motion with its region. Hence, we assign an occlusion boundary to the region of similar motion. The motion of a boundary [pink arrow of Fig. 7(c)] is the mean motion of the pixels on the boundary.

2) *Resolving the defocus ambiguity.* As shown in Fig. 8(a) and (b), the blurring degree of the frontmost yellow object is the same as that of the background. Hence, just from the degree of blur, the algorithm cannot tell whether an object is in front of or behind the focused area, but only the relative distance. We call this defocus ambiguity of depth from defocus.

To resolve the ambiguity, we use the focused region/object as the reference, and introduce pairwise occlusion relationship (a type of partial order) between regions/objects so as to sort their order in depth.

The depth reference region, r_0, is the one with the smallest degree of blur (i.e., with the minimum average $\tilde{\alpha}_d(p),\ p \in r_0$). The depth partial order of two adjacent regions r_{i_1} and r_{i_2} with a shared boundary b_i is decided by

$$\mathcal{PO}(r_{i_1}, r_{i_2}) = \begin{cases} r_{i_1} \prec r_{i_2}, & b_i \in r_{i_1} \text{ and } b_i \in OB \\ r_{i_1} \succ r_{i_2}, & b_i \in r_{i_2} \text{ and } b_i \in OB \\ r_{i_1} \sim r_{i_2}, & b_i \in TE \\ \text{undecided}, & \text{otherwise} \end{cases} \tag{17}$$

where OB and TE denote the occlusion boundary set and the texture edge set, respectively. $r_{i_1} \prec r_{i_2}$ means the depth of r_{i_1} is smaller than r_{i_2}. $r_{i_1} \sim r_{i_2}$ means the two regions are of a similar depth. Taking Fig. 8 as an example, the girl's region B is the reference, according to the boundary assignment (c) and (d), $B \prec C$, $B \succ A$, and $A \prec C$, hence, $A \prec B \prec C$.

All the regions with decided pairwise partial orders are linked to a ranking list, and this ranking transfers to the anchor superpixels of the regions. Superpixels within a region share similar depths, i.e., $s_{ij} \sim s_{ik}$, if $s_{ij}, s_{ik} \in r_i$. Then, the depth value of a superpixel s_i is computed as

$$\alpha_d(s_i) = \begin{cases} \tilde{\alpha}_d(s_0) + |\tilde{\alpha}_d(s_i) - \tilde{\alpha}_d(s_0)|, & s_0 \prec s_i \\ \tilde{\alpha}_d(s_0) - |\tilde{\alpha}_d(s_i) - \tilde{\alpha}_d(s_0)|, & s_0 \succ s_i \\ \tilde{\alpha}_d(s_i), & s_0 \sim s_i \end{cases} \tag{18}$$

where s_0 denotes a reference anchor superpixel in r_0. Equation (18) resolves the defocus ambiguity by flipping a superpixel to the correct side of the reference point according to the depth partial order and keeping their relative distance unchanged. Fig. 8(e) shows the resolved depth ambiguity in anchor superpixels. For other regions/superpixels that are not in the ranking list or not associated to any boundaries, the depth values are left to be decided in the depth fusion phase.

4) *Multicue Fusion:* Fusing the depth maps from the three cues consist of three phases, 1) aligning the depth from the three cues to the same range, 2) depth value assignment on common anchor superpixels, and 3) propagating the depth values of depth anchors to the other superpixels in a frame.

(a) *Depth alignment.* Since each cue returns the depth of its own range, and the three cues should predict the same depth values on the common anchor superpixels, we use these common anchors to align the depth values from different cues.

Denote $Y = (\alpha_a(cs_1), \ldots, \alpha_a(cs_n))$ as the reference depth vector from the common anchor superpixels of a reference cue. (In the 2-D-to-3-D conversion, any of the three cues can be the reference cue. In this paper, we use the aerial perspective cue as the reference.) n is the number of common anchor pixels on the depth maps from the three cues, and cs_i denotes a common *anchor superpixel*. $\alpha_a(cs_i)$ is defined in (3). Denote $X = (\alpha_*(cs_1), \ldots, \alpha_*(cs_n))$ as a to-be-aligned depth vector from another cue, where $* \in \{m, d\}$, defined in

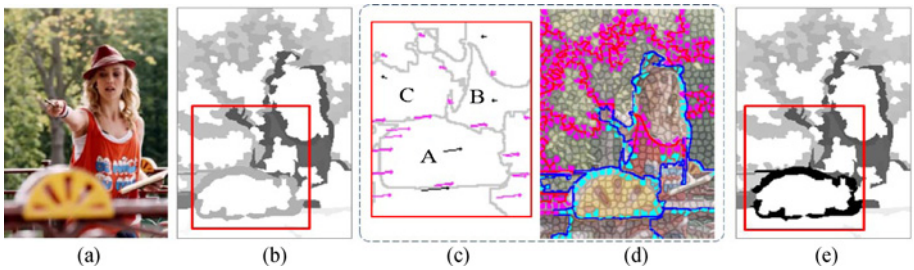

Fig. 8. Resolving the defocus ambiguity. (a) Video frame with defocus ambiguity. (b) Depth map computed using soft-max (15) in superpixels. The defocus ambiguity part is highlighted in a red rectangle. The darker the color of a pixel, the closer it is to the camera. (c) Motions of the image regions (black arrows) and boundaries (pink arrows). (d) Inferred TE (red) with their attached anchor superpixels (pink), and OB (blue) with their attached anchor superpixels (cyan). (e) Depth ambiguity is resolved by flipping the superpixels' depth values w.r.t. the reference region (the girl).

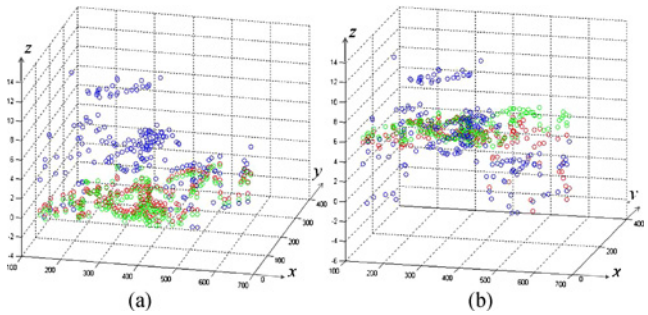

Fig. 9. Depth alignment. Each point is a common anchor superpixel of one of the three cues. The blue points are depth anchors from the reference aerial perspective cue, the green points from the motion cue and the red points from the defocus cue. (a) Before alignment. (b) After alignment.

(8) (18), respectively. A linear transformation is adopted to the depth alignment

$$aX + b = Y \tag{19}$$

where a and b are the transformation parameters, whose optimal values can be obtained by solving a least square problem. Fig. 9 shows the alignment result.

(b) *Depth Assignment on Common Anchors*. After alignment, on each common anchor superpixel, if the differences among the depth values of the three cues are within 10%, the depth value of the anchor is set to the arithmetic mean of the three depth values; Otherwise, the depth value of the anchor is taken as the one with the maximum confidence in (ν_a, ν_m, ν_o).

(c) *Depth Propagation*. For the noncommon-anchor superpixels, we estimate their initial fused depth values in the same way as the common anchors introduced above. Then, we have an initial depth map $\alpha_0(s)$. In this phase, the depth values on the common anchor superpixels are used as boundary condition and propagated to the rest of a frame subject to a smoothness prior and the initially fused depth constraint.

An optimal depth map α^* is inferred via the maximum a posterior (MAP) estimation on a MRF by minimizing the following energy function:

$$\alpha^* = \arg\min_{\alpha} E(\alpha|\alpha_0, I) \tag{20}$$

$$E(\alpha|\alpha_0, I) = E_u(\alpha, \alpha_0) + \lambda E_s(\alpha, I) \tag{21}$$

where E_u is the unary term, E_s is the pairwise term, and λ is the weighting factor to balance the two terms.

The unary term is defined as the cost on the deviation of the estimated depth $\alpha(s)$ from the initial value $\alpha_0(s)$ on each superpixel s,

$$E_u(\alpha, \alpha_0) = \sum_{s \in I} (\alpha(s) - \alpha_0(s))^2. \tag{22}$$

The pairwise term E_s is defined as

$$E_s(\alpha, I) = \sum_{s,t \in \partial} (\alpha(s) - \alpha(t))^2 e^{-\|I_s - I_t\|_2} \tag{23}$$

where neighbor superpixels, s and t, are assumed to have similar depth values if their colors are similar.

The optimization of $E(\alpha|\alpha_0, I)$ is solved by Graph Cuts [5], where the depth values are quantized to 15–25 levels and each level is considered as a label in the graph labeling problem. During the optimization the depth values on the common anchor superpixels are fixed. One example of the optimized depth map is shown in Fig. 5(f) and (g).

C. Foreground Depth Refinement

It is important to get accurate depth maps for key frames; not only does it ensure the quality of stereo visual effect, but also provides seeds for depth propagation to nonkey frames in a video shot. Although the above multicue fusion method can provide a good depth estimation for background, the depth values of foreground pixels can be inaccurate if the foreground moves fast. Moreover, cognitive studies [20] suggested that human visual system is more sensitive to foreground objects. Considering this, we add an interactive step for foreground depth refinement, which takes the following steps, 1) interactive segmentation of foreground objects, followed by 2) foreground depth reestimation and interactive adjustment.

1) *Foreground Segmentation:* The GrabCut [31] method is used to segment objects using user scribbles as shown in Fig. 3(b). In this paper, we modify the original GrabCut method by adding depth information to the model. In the original GrabCut method, the segmentation label $\ell_p \in \{0, 1\}$ for each pixel is obtained by iteratively minimizing an energy function

$$E(\ell, k, \theta, I) = U(\ell, k, \theta, I) + V(\ell, I) \tag{24}$$

where $U(\cdot)$ is the data term modeled by a Gaussian Mixture Model (GMM) of color with model parameter θ and k components, $V(\cdot)$ is the smoothness term, I is pixel color. (Please

refer to [31] for the details of the GrabCut method.) In our system, besides color, the depth values $\alpha(I)$ obtained from the multicue depth estimation module (described in Section IV-B) are also considered in both the data term and smoothness term. Hence, we modify the original GrabCut data term as

$$U(\ell, k, \theta, I, \alpha(I)) = \sum_{p} D(\ell_p, k_p, \theta, I(p), \alpha(p)) \tag{25}$$

where $D(\cdot) = -\lambda_c \log p(I(p)) - \lambda_\alpha \log p(\alpha(p))$. λ_c and λ_α are weights for the log probability of color and depth of GMM, respectively.

The modified smoothness term is

$$V(\ell, I, \alpha(I)) = \sum_{q \in \partial p} S(p, q) \tag{26}$$

where $S(\cdot) = [\ell_p \neq \ell_q] e^{-\gamma_1 ||p-q|| - \gamma_2 |\alpha(p)-\alpha(q)|}$, and $[\ell_p \neq \ell_q]$ equals 1 if $\ell_p \neq \ell_q$; otherwise 0. $||p - q||$ is the Euclidean distance between two pixel coordinates. $|\alpha(p) - \alpha(q)|$ is the absolute difference of the depth, which penalizes depth difference between neighbor pixels. In the future, we could automatically detect visual saliency regions[40] to reduce user interactions of object segmentation.

If the GrabCut does not give an accurate segmentation, the system provides users an interactive interface to adjust object contours. We adopt Intelligent Scissors [25] to facilitate users to refine the segmentation contours efficiently. Fig. 3(c) shows the refinement of the hat contour by just dragging the three blue control points using Intelligent Scissors.

2) *Foreground Depth Reestimation:* Foreground object depth and geometry estimation can be inaccurate if the objects are in fast motion or no defocus cue can be leveraged. If the multicue depth estimation result is not satisfactory for an foreground object, the system provides a user interface to reestimate a foreground object depth by allowing users to choose one of the three cues. A user can choose a cue either according to his/her experience by observing the video clip or by comparing the depth maps generated by the three cues and decide the best one for the object shape. For example, 1) if the defocus cue is obvious, user can select to use depth-from-defocus. An example is shown in Fig. 10(d). 2) If the object is static and the camera motion satisfies the SfM constraint, users can use SfM. 3) If there is not a good cue to use, users can just use a plane surface model to represent the object geometry as shown in Fig. 10(c).

Besides the geometry of foreground objects, their locations in the scene may also need adjustment. Although the foreground depth by multicue fusion may be noisy, we assume that the majority of pixels are reasonable; Hence, we can decide the object location/depth by majority voting. Or users can manually adjust foreground object locations using the 3-D labeling interface [shown in Fig. 3(a) Window 3].

D. Foreground Propagation

After the foreground depth is updated, the whole depth map for a key frame is completely generated. In this section, we introduce a method, which propagates the key frame depth to nonkey-frames within the same shot based on object tracking [39] and segmentation.

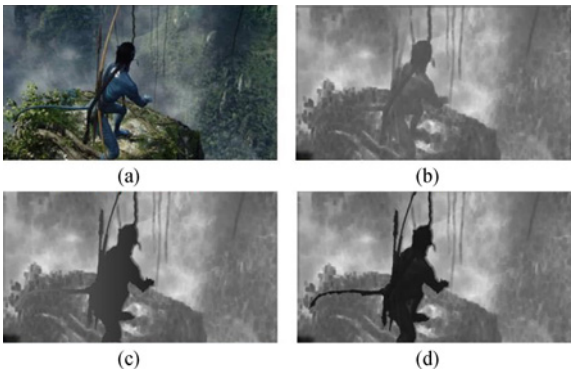

Fig. 10. Depth estimation and foreground depth reestimation.

Given an foreground object and the depth map of a key frame, we first directly warp the object contours to the nonkey-frames according to the object motion, and use the warped curve as initial object location. Then the warped contour is evolved to the object boundary using an adapted Level-set method based on [6]. We introduce the details as follows.

Let $\phi : \Omega \to \Re^2$ be a level set function defined on a domain Ω. Then the proposed energy functional $\xi(\phi)$ is

$$\xi(\phi) = \mu R_\varphi(\phi) + E_{img}(\phi) \tag{27}$$

where μ is a constant. The level set regularization term $R_\varphi(\phi)$ is

$$R_\varphi(\phi) = \int_\Omega \varphi(|\nabla\phi|)dp \tag{28}$$

where φ is a potential function $\varphi : [0, \infty) \to \Re$

$$\varphi(\phi) = \frac{1}{2}\int_\Omega (|\nabla\phi| - 1)^2 dp. \tag{29}$$

It is a metric to characterize how close a function $|\nabla\phi|$ is to a signed distance function, which satisfies a desirable property of $|\nabla\phi| = 1$ in $\Omega \to \Re^2$.

$E_{img}(\phi)$ is the adapted term of the external energy in [6]. It depends upon the image and its depth map

$$E_{img}(\phi) = aL_g(\phi) + bA_g(\phi) + c\Delta_\kappa(\phi) \tag{30}$$

where a, b and c are the coefficients of the energy functionals $L_g(\phi)$, $A_g(\phi)$ and $\Delta_d(\phi)$, respectively

$$L_g(\phi) = \int_\Omega g\delta(\phi)(|\nabla\phi|)dx \tag{31}$$

$$A_g(\phi) = \int_\Omega gH(-\phi)dx \tag{32}$$

and

$$\Delta_\kappa(\phi) = \int_\Omega \kappa H(-\phi)dx \tag{33}$$

where δ and H are the Dirac delta function and the Heaviside function, respectively. Function g is defined as an edge indicator by

$$g = \frac{1}{1 + |\nabla G_\sigma * I|^2} \tag{34}$$

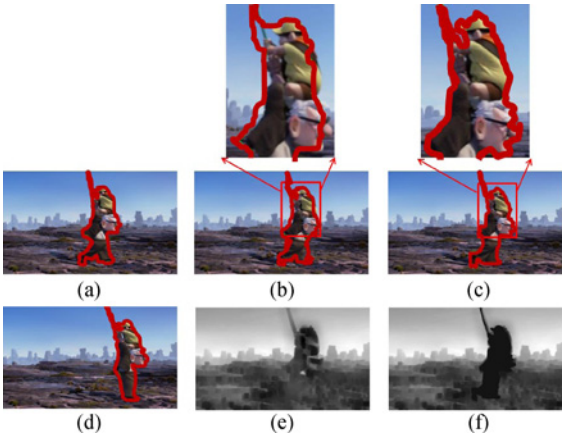

Fig. 11. Foreground object tracking and depth update. (a) Labeled foreground object (red curve denotes its boundary) at a key frame $t-1$. (b) Warped foreground object contour from frame $t-1$ to frame t using KLT [24]. (c) Object boundary refinement by the proposed Level-set method. (d) Tracked object at frame $t+1$. (e) Initial depth map estimated by multicue fusion at frame t. (f) Updated depth of foreground object after boundary refinement.

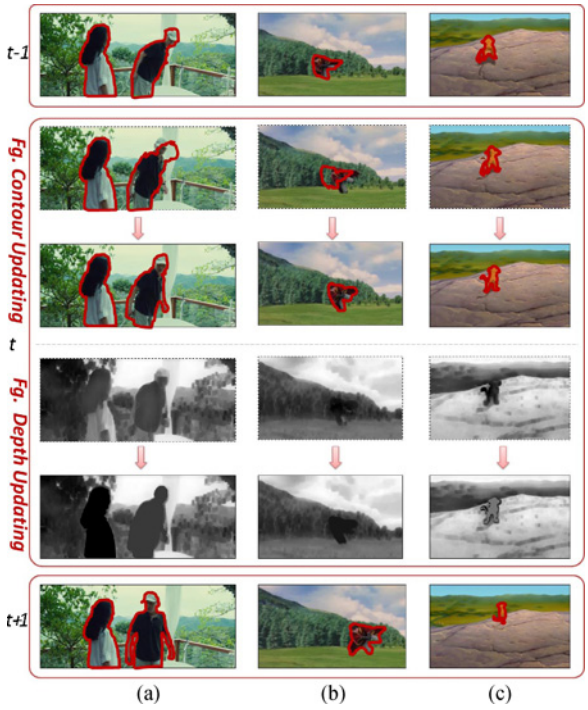

Fig. 12. Results for depth propagation in a video shot.

where I denotes the frame on the domain Ω, and G_σ is a Gaussian kernel with a standard deviation σ. $L_g(\phi)$ relates to the length of the zero level curve ϕ and $A_g(\phi)$ is introduced to accelerate the process of curve evolution.

In the proposed propagation method, we integrate depth into the level set data term, the function κ is defined as

$$\kappa = \frac{1}{1+\sum_{p,q\in N}|\alpha(p)-\alpha(q)|} \tag{35}$$

where N is a set of neighbor pixels including p and q in the domain Ω; $\alpha(p)$ is the depth value of pixel p.

The energy functional $\Delta_\kappa(\phi)$ is a new added term to the original formulation proposed in [6]. In addition to the intensity map, it is proposed to estimate accurate contours of foreground objects according to the gradient field of a depth map, assuming that pixels on an object share similar depths. $\Delta_\kappa(\phi)$ penalizes the case when the depth values inside the segmented object are quite different with each other. In experiments, it confirms that the final segmentation result is improved greatly by adding this term and the object's depth is more reliable after it is updated based on the refined segment, as shown in Fig. 11.

To speed up the evolution process, we initialize the level set function based on the segmentation result of its previous frame. Given the segmented foreground object, we extract speeded up robust features [4] as feature points for feature tracking and warp the contour using the KLT tracking method [24]. The approximated contour of the foreground object is propagated from its previous frame and can be used as a good initialization, which reduces the number of iterations to move the zero level set to the desired object boundary compared to a general initialization. We apply a standard method to minimize the energy functional by finding the steady state of its gradient flow as [3].

More propagation results are shown in Fig. 12. The first row shows the labeling at frame $t-1$. The warping results using KLT [24] are shown in the second row. The direct warping seems inaccurate, especially when the displacement of the foreground objects is large, e.g., Fig. 12(b). The third row displays the refined segmentation results by the proposed method. We can see that the contours of the objects are localized nicely. The fourth row shows the depth maps estimated by the multicue fusion method described in section IV-B). In the fifth row, updated/reestimated foreground depth of the objects are presented. The improvement is evident. The objects at frame $T+1$ can be reliably tracked from T using the same method.

E. Stereo View Frame Synthesis

Before stereo view synthesis, it is necessary to convert a depth map $\alpha(I)$ to a disparity map $d(I)$. The disparity value $d(p)$ at pixel p is the horizontal coordinate difference between the corresponding pixels in the left view and the right view. When a stereo frame is displayed, a pixel with negative disparity value is perceived as a point outward screen by viewers, and vice versa. A larger absolute value of $d(p)$ indicates a longer distance between the screen and the point.

In the system, a pixel depth $\alpha(p)$ is converted to a pixel disparity $d(p)$ by

$$d(p) = s\cdot W_I\cdot\left(\frac{\alpha(p)-\alpha_{min}(I)}{\alpha_{max}(I)-\alpha_{min}(I)}-\tau\right) \tag{36}$$

where W_I is the image width, $\alpha_{max}(I)$ is the maximal depth values in an image I. s is the control factor that restricts the maximal absolute disparity, and it makes the system adaptive to different screen sizes. Generally speaking, for devices whose screens are larger than 70 inches, s should be less than 1%. τ $(0 \le \tau < 1)$ is the parameter that shifts the disparity to a negative value and produces inward/outward

screen effects. Without τ, all the disparities will be positive, then the outward screen effect cannot be rendered. In our system, τ is determined by the stereo effect of a reference foreground object in the scene. Let p_{ref} be the reference point on a foreground object, and $\alpha(p_{\text{ref}})$ be its depth value. τ is computed as

$$\tau = \frac{\alpha(p_{\text{ref}}) - \alpha_{min}(I)}{\alpha_{max}(I) - \alpha_{min}(I)} - \frac{d(p_{\text{ref}})}{s \cdot W_I} \tag{37}$$

$d(p_{\text{ref}})$ is the disparity of the reference point/object, which is automatically predicted by a trained disparity estimation model—a multilabel support vector machine (SVM).

We use motion and position of a segmented object as the features to predict its disparity value. The feature vector includes four components: 1) object motion magnitude and orientation histograms, 2) pixel location histogram of the object region, 3) the mean and variance of the depth values (by multicue estimation in Section IV-B) of the object pixels, and 4) the motion magnitude and orientation histograms of the background region, which is an indication of camera motion.

In the learning phase, given a pair of training stereoscopic video sequences sampled from commercial 3-D movies, disparity maps are directly computed by a state-of-the-art stereo matching method [33]. Notice that we do not perform parallel view rectification before the stereo matching, hence, the disparities are signed values. Since the disparity values obtained by the stereo matching [33] are quantized into several discrete disparity levels, we use them as the depth labels of the pixels in the multilabel SVMs (i.e., each disparity value of a pixel is considered as a class label in the SVMs). After the motion feature and position feature are extracted, the SVM is trained in the one-vs-all manner, which means that data of the target class are positive examples, and all the rest of the data are considered as negative examples. It is expected that the trained model can capture the correlation between the motion and the signed disparity. In the testing phase, features are first extracted from the test 2-D video, then object disparities are predicted.

When synthesizing the stereo view, an original 2-D frame I is considered as the middle view between the synthesized left view I_l and right view I_r of the stereo image pair. So, I_l and I_r can be synthesized by warping I according to the predicted disparity map $d(I)$

$$I_r(p) = I(p + 0.5 \times d(p)) \tag{38}$$

$$I_l(p) = I(p - 0.5 \times d(p)). \tag{39}$$

After warping, there appear some "holes" due to the discontinuity of the disparities. An inpainting method [34] is utilized to fill these holes and the system obtains the final stereo views.

To make the scene depth configuration consistent between similar shots, the system passes parameters to control the depth range and disparity range of a scene.

V. Experiments and Results

In experiments, we convert several well-known films into stereo. Fig. 14 shows some converted key frames from three movies. We evaluate the system in both conversion quality and conversion efficiency.

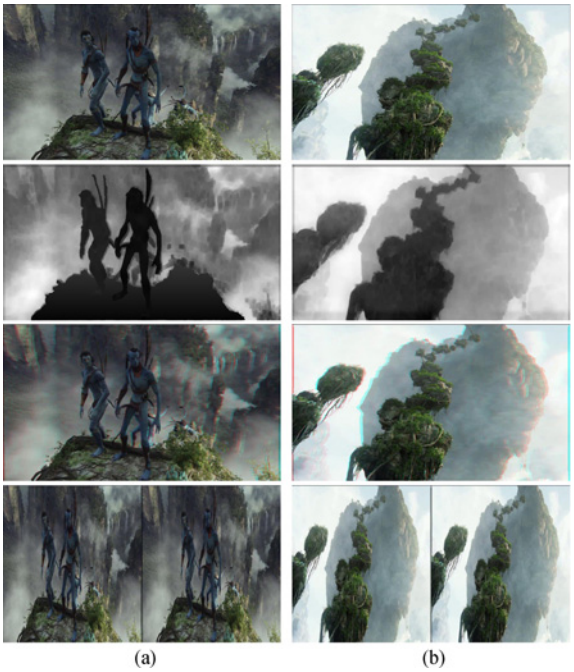

Fig. 13. Example key frames and their conversion results. First row: two original frames from *Avatar*. Second row: depth maps generated by the proposed system (left foreground with user interaction; right: no interaction). Third row: converted anaglyph. Forth row: the converted stereo frames (side-by-side).

A. Video Sets for Quality Evaluation

We prepare three sets of videos for quality evaluation. 1) Video Set I—the anchor set, which contains five stereo video clips from five original stereo movies. Two of them are synthetic/cartoon movies (labeled with "*" in Table I) and the other three are of real scenes. 2) Video Set II—synthesized stereo clips by the proposed system (side-by-side format). The clips are synthesized from the left view videos in Video Set I (Two examples are shown in Fig. 13). For the impairment assessment of video quality, different from (39), here we synthesize the right view as $I_r(p) = I(p + d(p))$ ($d(p)$ is defined in (36). Fifteen graduate students were recruited and trained to use the proposed system. They are divided into five groups and each group is asked to convert one clip into stereo within a given time (3–10 in according to the length of the clips and complexity of the scenes). Thus, there are three versions for each of the five clips so as to diminish the individual performance effect on the conversion quality. 3) Video Set III—synthesized stereo clips by the system proposed by Liao *et al.* [22]. Another 15 students are trained to use the system [22] to generate five groups of stereo video clips (each clip has three versions). The right views are synthesized from the left views of Video Set I also according to $I_r(p) = I(p + d(p))$.

B. Conversion Quality Evaluation

The quality of the converted videos is evaluated subjectively. Fifteen subjects (they are not the students who converted the

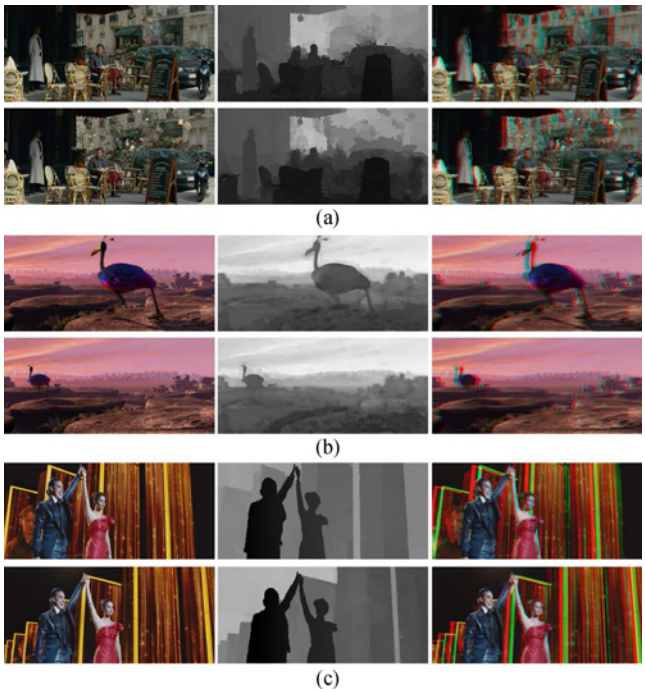

Fig. 14. Example results on three video sequences, (a) *Inception*, (b) *Up*, and (c) *Hunger Games*. First column: two key frames from each of the three original videos. Second column: depth maps. Third column: Red-cyan anaglyphs .

TABLE I
COMPARISON OF CONVERTED VIDEO QUALITY OF THREE METHODS WITH EVALUATION SCORES (VARIANCE)/ACTUAL AVERAGE TIME COST

	Our Method	The method in [22]	Samsung
clip 1	**4.39** (0.39)/270 s	4.20 (0.44)/352 s	3.52 (0.41)
clip 2*	4.72 (0.28)/202 s	**4.75** (0.36)/275 s	4.05 (0.43)
clip 3	**4.65** (0.30)/335 s	4.60 (0.42)/478 s	3.76 (0.39)
clip 4*	**4.63** (0.36)/230 s	4.42 (0.40)/301 s	3.91 (0.37)
clip 5	**4.60** (0.34)/187 s	4.48 (0.32)/230 s	3.64 (0.44)

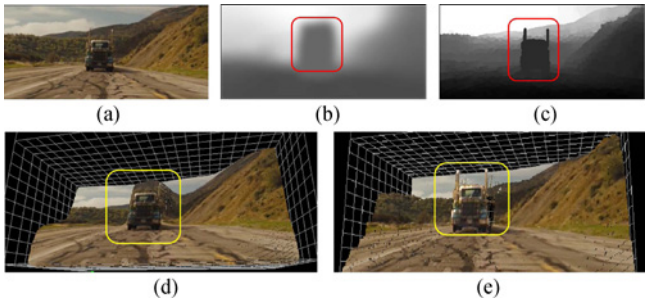

Fig. 15. Comparison results between the method in [22] and the proposed method. (a) Original 2-D frame. (b) and (d) Depth map generated using the method in [22] and its 3-D visualization with texture mapping. (c) and (e) Depth map generated using the proposed method and its 3-D visualization.

videos, and not familiar with the videos) are asked to compare the stereo effects between the converted videos and the anchors. All the subjects are with normal or rectified normal visual acuity and normal stereoscopic acuity. (However, we did not test their eye dominance.) The degradation category rating from ITU-T Recommendation P.910 [29] is used in the subjective experiments. Samsung Active Shutter 3DTVs (UA55ES6100) are used to display stereo videos with a $2m$ viewing distance in a dark room.

Subjects are asked to give a score to a converted video clip (in real values) by comparing it with the corresponding anchor clip according to a five-level impairment scale. The score ranges from 1 to 5; 5 indicates the impairments of converted video "are imperceptible," and 1 means the "impairments are very annoying." As shown in Table I, average evaluation scores, variances and conversion time of the proposed method and another two systems are recorded in the experiments.

Experiment 1—stereo quality evaluation of our system We evaluate the stereo qualities of the converted video clips and their key frames.

Seventy-five key frames are sampled from the converted videos (Video Set II). Each key frame is the median frame of a video shot according to the shot segmentation results introduced in Section IV-A. Subjects give scores when they are presented with the converted key frames and their corresponding anchor frames (in Video Set I). The experiments report an average score, 4.40 (with a variance of 0.42). Most of time the subjects could not perceive the difference in the background. This supports our assumption that a better foreground will greatly improve visual impression.

For each anchor stereo clip in Video Set I, the 15 subjects compare it with the three versions of converted videos in Video Set II, and then report totally 45 subjective scores (three versions of converted videos $\times$ 15 observers). The first column in Table I shows the average scores (with variances) of the converted videos generated by the proposed system.

The average score of all converted videos increases to 4.59 compared with the score of 4.40 on the key frames. This indicates that while watching videos, the artifacts are less noticeable.

Experiment 2—comparison with the interactive conversion system proposed in [22] The converted video clips by [22] in Video Set III are also compared with their original stereo clips in Video Set I. Their evaluation scores of visual quality reported by the subjects are recorded. The average score of each clip is shown in the second column of Table I. Comparing the first two columns, we can see that the proposed system achieves a better or comparable conversion quality than the system proposed by [22]. It can be noted that the users spent less average actual conversion time on each clip using our system.

We also compare the depth maps generated by the two systems in Fig. 15. The depth estimation in [22] requires propagating the depth values from feature point pixels to the other pixels. Since the propagation is based on the smoothness assumption in space and time but without occlusion boundary constraint, the region boundaries in the depth maps tend to be defused. Fig. 15(b) shows a depth map estimated by [22]. It can be clearly seen that the foreground region (in the red box) is enlarged. When the scene is rendered in 3-D according to the depth map [Fig. 15(d)], the background geometry is distorted. In contrast, the depth map [Fig. 15(c)] computed by

TABLE II
TIME CONSUMPTION OF OUR SYSTEM

Item	Time(per frame)
Shot segmentation	10 ms
Depth propagation	3 s
Automatic depth estimation	1.5 s
Depth to disparity conversion	0.1 s
Foreground segmentation	3 s to 5 s
Total	8 s to 10 s

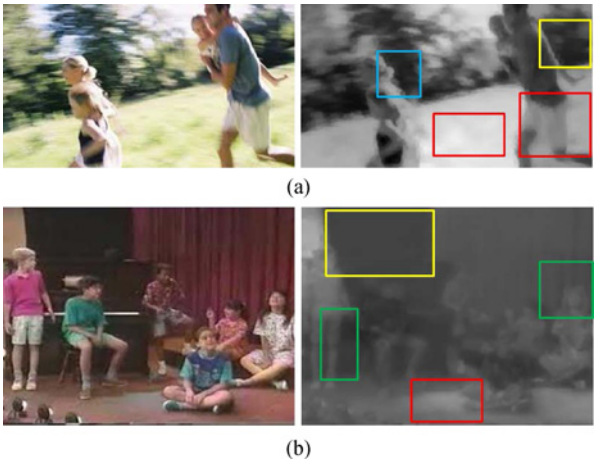

Fig. 16. Failure examples of the proposed method on low quality videos. Left column: two low quality video frames captured by nonprofessional cameras. The upper one has severe motion blur. The lower is low resolution. Right column: the depth maps generated by the proposed system without user interaction. The colored boxes highlight typical failure cases. Red boxes: wrongly estimated depth in bright regions at a close distance; Yellow boxes: wrongly estimated depth in dark regions at a far distance; Blue box: wrong depth orders in blurred regions. Green Boxes: wrong depth orders in stationary regions.

the proposed system is more precise and it renders a better result as shown in Fig. 15(d).

Experiment 3—comparison with an automatic conversion system. In this experiment, we compare the conversion quality to an automatic commercial conversion system—the 2-D-to-3-D conversion module in the Samsung 3DTV. The left views of the five stereo clips are input to the 3DTV in the 2-D-to-3-D mode. The 3DTV automatically synthesizes stereo videos. The subjects are asked score the stereo quality of the synthesized videos by comparing to the original stereo videos. As shown in Table I, the proposed interactive system obtains an obviously better conversion quality than the Samsung 3DTV.

C. Conversion Efficiency

The system is efficient in both interactive operations and the automatic modules. As shown in Table II, for a video with 1280×720 resolution, the average conversion time per frame is about 8 to 10 s, among which user interaction costs about 3–5 s on a key frame, and the automatic computation takes about 5 s per frame, of which the depth estimation (1.5 s) can be computed off-line before interactions.

We compare the key components in our system and the system in [22] particularly of the interaction modules. In our system, the interaction is to segment foreground objects. In [22], the interaction is to label the depth difference between regions. According to user feedback, object segmentation is more intuitive and takes less time than labeling of depth difference between regions.

The IMAX system is a commercial system of movie production. From media reports, the IMAX system takes about 6 to 10 weeks to convert a 2-h 2-D film into a stereo one; James Cameron used 60 weeks and 750 000 man hours to convert the Titanic[15], while our system only takes about 12 days of PC-hours for automatic conversion and 20-50 man-hours for user interaction.

VI. CONCLUSION

We presented an interactive system of 2-D-to-3-D video conversion, which is comprehensive and consists of a number of modules ranging from depth estimation, depth-to-disparity conversion, stereo view synthesis, to video coding/decoding. The proposed depth estimation method fuses three cues that govern different depth regimes from close to far. The optimal depth is inferred by minimizing the energy defined on a Markov Random Field. Experiment results demonstrate the advantage of the proposed system.

However, there are some limitations in this system. The reliability of the automatic depth estimation using the three depth cues depends on the quality of videos, i.e., the method is only good at converting videos with high resolution and strong depth cues. Given a degraded video frame shot by a low resolution nonprofessional camcorder as shown in Fig. 16, the three cues are usually hard or unreliable to extract, then the depth estimation will be rather unsatisfactory. For example, in Fig. 16(a), there is little defocus cue, the motion blur ruins accurate motion estimation, and only aerial perspective can be applied. As mentioned before (in Section IV-B), due to the algorithmic flaws in each monocular cue, the estimated depth map contains a lot of errors. Similarly, Fig. 16(b) shows a low-resolution frame of indoor scene almost without any of the three cues. The estimated depth map is erroneous.

In addition, the integration of monocular cues is still naive, e.g., the alignment of depth maps of different cues is a linear model. In the future, we shall extend the current system by incorporating more monocular cues for depth estimation and study advanced models of depth alignment, so that it is able to robustly estimate scene depth from regular or even low quality videos as well.

REFERENCES

[1] (2011). H.264/AVC Joint Model 18.0 (JM-18.0) [Online]. Available: http://iphome.hhi.de/suehring/tml/download.
[2] R. Achanta, A. Shaj, K. Smith, A. Lucchi, P. Fua, and S. S(r)¹sstrun, "Slic superpixels compared to state-of-the-art superpixel methods," in *Proc. PAMI*, 2012, vol. 34, no. 11, pp. 2274–2481.
[3] G. Aubert and P. Kornprobst," *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations* (Applied Mathematics Science 147), Berlin, Germany: Springer, 2002.
[4] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *Proc. ECCV*, 2006, pp. 404–417.

[5] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimisation via graph cuts," *IEEE Trans. PAMI*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
[6] L. Chunming, X. Chenyang, G. Changfeng, and M. D. Fox, "Distance regularized level set evolution and its application to image segmentation," *IEEE Trans. Image Process.*, vol. 19, no. 12, pp. 3243–3254, Dec. 2010.
[7] A. Criminisi, I. Reid, and A. Zisserman, "Single view metrology," in *Proc. IJCV*, 2000, vol. 40, no. 2, pp. 123–148.
[8] P. Favaro, S. Oshe, S. Soatto, and L. Vese, "3D shape from anisotropic diffusion," in *Proc. IEEE Int. Conf. CVPR*, Jun. 2003, pp. 179–186.
[9] G. Guo, L. Liu, Z. Zhang, Y. Wang, and W. Gao, "An interactive method for curve extraction," in *Proc. ICIP*, 2010, pp. 1905–1908.
[10] M. Guttmann, L. Wolf, and D. Cohen-Or, "SEmi-automatic stereo extraction from video footage," in *Proc. ICCV*, 2009, pp. 136–142.
[11] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conf.*, 1988, pp. 147–151.
[12] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge Univ. Press, 2000.
[13] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," in *Proc. CVPR*, 2009, pp. 2341–2353.
[14] D. Hoiem, A. A. Efros, and M. Hebert, "Closing the loop on scene interpretation," in *Proc. CVPR*, 2008.
[15] (2012). IEEE and the Titanic: A Century of Technological Heritage and Innovation [Online]. Available: http://www.ieee.org/about/news/2012.
[16] [Online]. Available: http://www.imax.com/corporate/technology/2d-to-3d-conversion/
[17] D. Kim, D. Min, and K. Sohn, "A stereoscopic video generation method using stereoscopic display characterization and motion analysis," *IEEE Trans. Broadcasting*, vol. 54, no. 2, pp. 188–197, Jun. 2008.
[18] S. Knorr and T. Sikora, "An image-based rendering IBR approach for realistic stereo view synthesis of TV broadcast based on structure from motion," in *Proc. ICIP*, 2007.
[19] J. J. Koenderink, "The structure of images," *Biol. Cybern.*, vol. 50, no. 5, pp. 363–370, 1984.
[20] J. J. Koenderink, A. J. van Doorn, A. M. Kappers, and J. T. Todd, "Ambiguity and the 'mental eye' in pictorial relief," *Perception*, vol. 30, no. 4, pp. 431–482, 2001.
[21] I. Koprinska and S. Carrato, "Temporal video segmentation: A survey," *Signal Process.: Image Commun.*, vol. 16, no. 5, pp. 477–500, Jan. 2001.
[22] M. Liao, J. Gao, R. Yang, and M. Gong, "Video stereolization: Combining motion analysis with user interaction," *IEEE. Trans. TVCG*, vol. 18, no. 7, pp. 1079–1088, Jun. 2011.
[23] C. Liu, H. Liu, S. Jiang, Q. Huang, Y. Zheng, and W. Zhang, "JDL at TRECVID 2006 shot boundary detection," in *Proc. TRECVID*, 2006, pp. 1–4.
[24] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. IJCAI*, 1981, pp. 121–130.
[25] E. Mortensen and W. Barrett, "Intelligent scissors for image composition," in *Proc. SIGGRAPH*, 1995, pp. 191–198.
[26] K. Moustakas, D. Tzovaras, and M. Strintzis, "Stereoscopic video generation based on efficient layered structure and motion estimation from a monoscopic image sequence," *IEEE Trans. CSVT*, vol. 15, no. 8, pp. 1065–1073, Aug. 2005.
[27] V. P. Namboodiri and S. Chaudhuri, "Recovery of relative depth from a single observation using an uncalibrated cCamera," in *Proc. CVPR*, 2008, pp. 1–6.
[28] A. P. Pentland, "A new sense for depth of field," in *Proc. PAMI*, 1987, vol. 9, no. 4, pp. 523–531.
[29] ITU-T Recommendation P.910, "Subjective video auality assessment methods for multimedia applications," International Telecommunication Union, 2008.
[30] E. Rotem, K. Wolowelsky, and D. Pelz, "Automatic video to stereoscopic video conversion," in *Proc. SPIE*, 2005, vol. 5664, pp. 198–206.
[31] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut—interactive foreground extraction using iterated graph Cuts," in *Proc. SIGGRAPH*, 2004, pp. 309–314.
[32] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3-D scene structure from a single still image," *IEEE Trans. PAMI*, vol. 31, no. 5, pp. 824–840, May 2008.
[33] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithm," in *IJCV*, 2002, vol. 47, nos. 1–3, pp. 7–42.
[34] A. Telea, "An image inpainting technique based on the fast marching method," *J. Graphics, GPU, Game Tools*, vol. 9, no. 1, pp. 23–34, 2004.
[35] J. Tighe and S. Lazebnik, "Superparsing: Scalable nonparametric image parsing with superpixels," in *Proc. ECCV*, 2010, pp. 352–365.
[36] C. Tomasi, "Shape and motion from image streams under orthography: A factorization method," in *IJCV*, 1992, vol. 9, no. 2, pp. 137–154.
[37] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large dataset for non-parametric object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1958–1970, Nov. 2008.
[38] S. A. Valencia and R. M. Rodriguez-Dagnino, "Synthesizing stereo 3D views from focus cues in monoscopic 2D images," in *Proc. SPIE*, 2003, vol. 5006, pp. 377–388.
[39] Y. Wang and S. C. Zhu, "Modeling complex motion by tracking and editing hidden Markov graphs," in *Proc. CVPR*, 2004, pp. 856–863.
[40] W. Wang, Y. Wang, Q. Huang, and W. Zhu, "Measuring visual saliency by site entropy rate," in *Proc. CVPR*, 2010, pp. 2368–2375.
[41] Z. Zhang, Y. Wang, T. Jiang, and W. Gao, "Stereoscopic learning for disparity estimation," in *Proc. ISCAS*, 2011, pp. 365–368.
[42] Z. Zhang, Y. Wang, T. Jiang, and W. Gao, "Visual pertinent 2D-to-3D video conversion by multi-cue fusion," in *Proc. ICIP*, 2011, pp. 909–912.
[43] Z. Zhang, C. Zhou, B. Xin, Y. Wang, and W. Gao, "An interactive system of stereoscopic video conversion," in *Proc. ACM MM*, 2012, pp. 149–158.

Zhebin Zhang received the B.S. degree from the Department of Computer Science and Technology, the Beijing University of Posts and Telecommunications, Beijing, China in 2005, and the Ph.D. degree in computer sciences from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2012.

He is a currently a Post-Doctoral Researcher with the School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His current research interests include computer vision, image, and video process.

Chen Zhou received the B.S. degree from Computer Science Department, Peking University, Beijing, China in 2011, and is currently pursuing Ph.D. degree with the Computer Science of Peking University, Beijing, China.

His current research interests include computer vision, especially 3-D reconstruction.

Yizhou Wang received the bachelor's degree in electrical engineering from Tsinghua University, Beijing, China in 1996, and the Ph.D. degree in computer science from the University of California, Los Angeles in 2005.

He is a Professor of Computer Science Department at Peking University (PKU), Beijing, China. He is a Vice Director of the Institute of Digital Media, PKU, and the Director of New Media Lab of the National Engineering Lab of Video Technology. He was a member of Research Staff of Palo Alto Research Center of Xerox from 2005 to 2008. His current research interests include computer vision, statistical modeling and learning, and digital visual arts.

Wen Gao (F'13) received the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He is currently a Professor with the Peking University, Beijing, China, since 2006. He was with the Harbin Institute of Technology from 1991 to 1995, as Professor, Chairman of Department of Computer Science. He was also with the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), as Professor, from 1996 to 2005. During his career at CAS, he served as the Managing Director of ICT from 1998 to 1999, the Executive Vice President of Graduate School of Chinese Academy of Sciences from 2000 to 2004, the Vice President of the University of Science and Technology China from 2000 to 2003. His current research interests include in the areas of video coding and processing, face recognition, image retrieval, and multimodal interface.

Dr. Gao is a member of the Chinese Academy of Engineering.