An Interactive System of Stereoscopic Video Conversion

Zhebin Zhang[†] Chen Zhou[†] Bo Xin[†] Yizhou Wang[†‡] Wen Gao[†‡]
[†]National Engineering Lab. for Video Technology,
[‡]Key Lab. of Machine Perception (MoE),
School of EECS, Peking University, Beijing, 100871, China
{zbzhang, zhouch, boxin, yizhou.wang, wgao}@pku.edu.cn

ABSTRACT

With the recent booming of 3DTV industry, more and more stereoscopic videos are demanded by the market. This paper presents a system of converting conventional monocular videos to stereoscopic ones. In this system, an input video is firstly segmented into shots to reduce operations on similar frames. Then, automatic depth estimation and interactive image segmentation are integrated to obtain depth maps and foreground/background segments on selected key frames. Within each video shot, such results are propagated from key frames to non-key frames. Combined with a depth-to-disparity conversion method, the system synthesizes the counterpart (either left or right) view for stereoscopic display by warping the original frame according to disparity maps. For evaluation, we use human labeled depth map as the reference and compute both the mean opinion score (MOS) and Peak signal-to-noise ratio (PSNR) to evaluate the converted video quality. Experiment results demonstrate that the proposed conversion system and methods achieves encouraging performance.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Sensor fusion, Depth cues*

Keywords

3DTV, 2D-to-3D conversion, depth estimation, depth to disparity, foreground segmentation

1. INTRODUCTION

Due to the amazing development speed of 3DTV industry, e.g. broadcasting of many 3D 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 stereoscopic cameras. Converting conventional monocular videos into stereoscopic ones is definitely a complimentary solution.

MM'12, October 29–November 2, 2012, Nara, Japan.

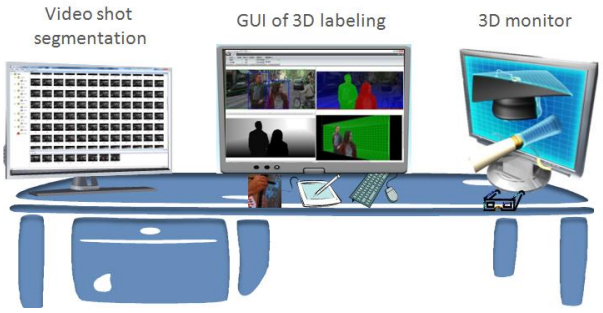

Figure 1: The interfaces of the stereoscopic conversion system.

However, automatic 2D-to-3D video conversion still remains a challenging problem. This is mainly because that the core problem – depth from monocular view – has not been solved. Although there exist techniques to estimate depth from monocular image sequences, such as structure from motion (SfM) [28], it requires such a constrained camera motion and foreground stillness and rigidity assumptions that a large proportion of real videos falls beyond its regime. In addition, in order to obtain good estimate, these algorithms always require high quality intermediate results such as good correspondence matching and motion estimation. Whereas, in real videos, such requirement may be too demanding due to all kinds of variations and occlusions.

In this paper, an interactive system is proposed to convert monocular videos into stereoscopic ones (shown in Figure 1). We introduce human in the loop only to rescue the deficiency of the state-of-the-art algorithms. In the system, advanced video processing techniques are embedded so as to increase the conversion efficiency. Specifically, before the conversion, the system segments an input monocular video into shots so that the labeled information such as foreground object contours and depths can be propagated efficiently and reliably within each shot. We propose a new multi-cue depth estimation method by integrating a few robust monocular depth perception cues adopted by human beings, such as depth from defocus, depth from aerial perspective, depth from motion. These cues largely improve the depth estimation performance by freeing the constraints of the structure from motion (SfM) method and its extensions (e.g. [16],[28]). Using the proposed method, the estimated

depth is continuous and its range is much wider than SfM based methods. To improve the stereoscopic visual quality, we adopt interactive methods to segment foreground objects, and estimate their shape and depth position in particular. We propose to integrate depth information in tracking algorithms so as to improve the foreground object tracking accuracy within each shot. Moreover, a depth-to-disparity conversion model based on supervised learning is proposed, which predicts foreground disparity according to its motion, screen location and the background motion. The disparities of the rest pixels are computed according the their relative depth w.r.t. the foreground object. This method automatic generates inside/on/outside screen visual effect learned from stereoscopic movies.

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

Another contribution of the paper is that, due to the lack of evaluation methods in this field (especially on conventional movies), we propose one using human labeled depth maps as reference so as to assess the visual quality of converted stereoscopic videos both subjectively and objectively. Specifically, we compute both the mean opinion score (MOS) and Peak signal-to-noise ratio (PSNR) to evaluate the converted video quality. The experiment results demonstrate the advantage of the proposed system.

The rest of the paper is organized as follows: some closely related work is introduced in Section 2. In Section 3, we describe the work flow and architecture of the system, followed by expatiations of important functions and methods in Section 4. In section 5, evaluation and experimental results are provided. Finally, Section 6 concludes the paper.

2. RELATED WORK

In the literature, the 2D-to-3D conversion methods can be roughly categorized into two classes, the automatic conversion and interactive conversion. Automatic conversion methods (e.g. [12][13][20][22]) exploit motion cues to predict the depth/disparity of pixels. Some commercial softwares, such as DDD's TriDef 3D player and Samsung's 3DTV, leverage both motion and image location priors to generate stereoscopic views from monocular videos. However, the motion cues, e.g. 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, specific image location assumptions 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 give an accurate estimation of depth/disparities of a scene given the current state of the art computer vision algorithms.

The other category of conversion methods exploit user interactions. For example, in [6] and [16], at key frames, user scribbles are used to initialize depth values and depth layers of objects in a scene respectively, and then the depth information is automatically propagated to non-key-frames. Compared with these methods, the proposed model integrates more monocular cues in depth estimation, so that it adapts to more variety real videos and generate a wider estimation of continuous depth range. IMAX developed a sophisticated interactive conversion commercial system [11], which requires intensive manual work and can generate impressive stereoscopic visual effect.

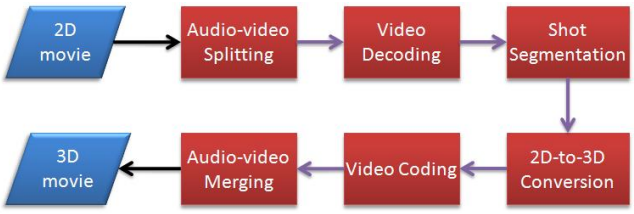

Figure 2: The work flow of the conversion system.

Estimating pixel disparities/depth from monocular videos is the key step in 2D-to-3D video conversion and it is an active yet challenging research topic in computer vision. Researchers took advantage of various cues for this task, such as photometry cues, e.g. [8][21][29], geometry cues, e.g. [4][5], motion cues, e.g. [12][13][20][22] and appearance cues, e.g. [6][24]. In the following we review these work even though some of them has not been applied to stereoscopic video conversion yet.

Photometry cues Objects in an image usually are not all in focus [21]. Valencia *et. al.* [29] used wavelet analysis and edge de-focus estimation to obtain relative depth. 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.* [8] proposed a dark channel prior to remove haze and also provided estimated depth map of a scene.

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 [4][5][7].

Motion cues Under the condition of constrained camera motion and assuming that scenes are static, there are two ways to estimate disparity maps, using (i) Structure from Motion (SfM), e.g. [28], and (ii) motion parallax [12]. However, in real scenario 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 in disparity estimation.

Appearance cues By using appearance features from superpixels (e.g. color, texture and shape), Hoiem *et. al.* [9] casted the depth estimation from single images to a multi-label classification problem. Saxena *et. al.* [24] employed Markov random field (MRF) to model the unstructured scene and directly learned the relation between the planar 3D structures and the superpixel's texture and color features. Both methods adopted supervised learning strategy, which requires training data to learn the model; this makes them difficult to be generalized to real data beyond the distribution of the training data sets.

3. SYSTEM OVERVIEW

As shown in Figure 2, the whole system starts with splitting video and audio apart, followed by video decoding, and then it converts the video into a stereoscopic one. After this, it compresses the video by the same codec, and finally the

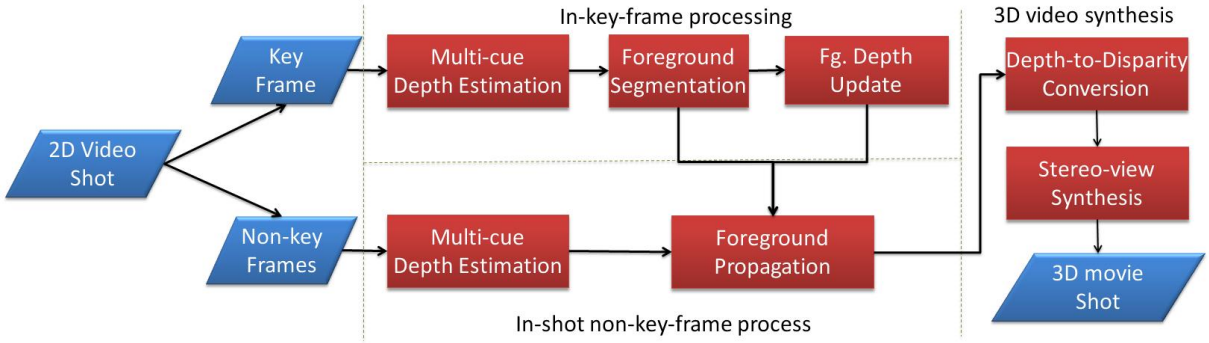

Figure 3: The flowchart of the *2D-to-3D conversion module* in Figure 2.

system merges the coded video with the original audio into a required video file format. The system is designed to reduce interactive operations as far as possible. Hence, before a video is sent to the 2D-to-3D conversion module, the whole video is segmented into shots so that the labeled information can be reliably propagated within a shot (Figure 5).

3.1 The interfaces and interactions

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

A shot segmentation interface (shown in Figure 5), where users can select certain shots to process.

A 3D labeling interface consists of four windows as shown in Figure 4 (a). In window 1, 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 estimated depth map of a frame. In window 3, both foreground objects and background are displayed in a 3D grid so that the user can clearly see their relative depth in 3D 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 inside, on or outside the screen). The depth value of every pixel of a scene as well as the virtual screen are manually adjustable, so the user can easily correct depth estimation errors, and acquire desired rendering effect w.r.t. the screen. Window 4 shows a color map demonstrating the inside/on/outside screen distribution of the objects in a scene. Red areas indicate that the regions are outside of the screen, green is on the screen, and blue is inside the screen.

A stereoscopic monitor, where users can examine the rendered stereoscopic effect of converted key frames and videos. If the results are not satisfying, users can refine it by adjusting the labels through the 3D labeling interface.

4. 2D-TO-3D CONVERSION METHODS

The proposed system first segments an input video into shots, then converts each shot into stereoscopic one with in following steps (shown in Figure 3): (i) An initial depth map for each frame is automatically predicted by a proposed multi-cue depth estimation method. (ii) In order to refine the initial depth map, for key frames, foreground objects are interactively labeled and segmented out from background. Consequently, the depth values in the foreground regions are updated. (iii) The foreground/ background labels are propagated to other frames and their initial depth maps are

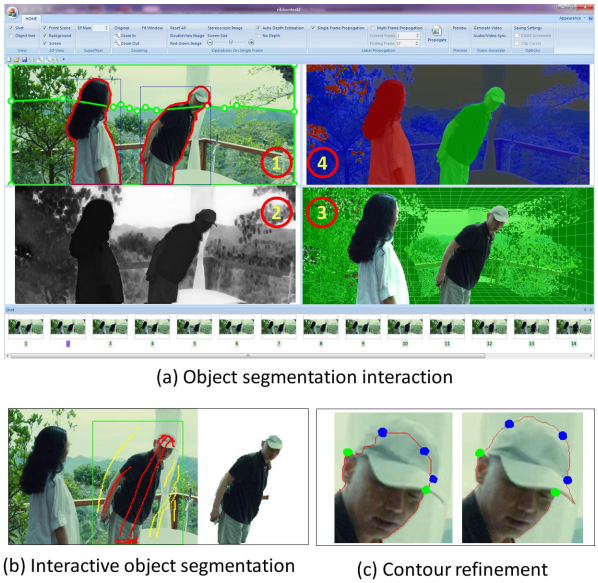

Figure 4: Segmentation of foreground objects.

also updated in the foreground regions; (iv) A disparity map for each frame is predicted using trained models. (v) The stereo view of the frame is synthesized based on the original frame and the disparity map.

In the rest of the section, we introduce the details of each step.

4.1 Shot segmentation

Video shots are defined as a set of meaningful and manageable segments which share the same background setting [15]. Consequently, information can be easily propagated within a shot. Our video segmentation algorithm follows the following steps: (i) feature extraction, (ii) dissimilarity computation between frames and (iii) shot boundary detection.

Feature extraction Both color histogram and tiny image [27] are used as features. In detail, a color histogram in RGB space with 16 bins for each channel is computed from a frame. To better preserve structural information, a frame is down-sampled by 16 times resulting in a 24×12 pixel tiny image.

Figure 5: Video shot segmentation interface.

Dissimilarity computation In [17], 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 between them. A similar definition of dissimilarity metric is proposed in this paper, except that the contribution of tiny image is taken into consideration.

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 $\Gamma_T(i,j) = \sum_{k=1}^{K_T} |T_i(k) - T_j(k)|$, respectively. The mutual information between the ith and the jth frame is computed based on color, $MI(C_i, C_j) = H(C_i) + H(C_j) - H(C_i, C_j)$.

Finally, the dissimilarity between two frames is

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

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,

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

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

4.2 Depth estimation by multi-cue fusion

In this paper, we deliberately select three depth perception cues to estimate the initial depth map of video frames, *i.e. motion cue, defocus cue and aerial perspective cue*, because each of them governs a different range of depth estimation. Using SfM method is able to accurately estimation scene depth at near distance; defocus cue is good to predict mid-range depth; and aerial perspective cue can give 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 (for example, motion cue is not applicable to textureless regions, a bright 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 robustly estimate scene depth by compensating the weakness of each other.

Denote the depth map of a frame I predicted by each cue as $\alpha_m(I)$, $\alpha_d(I)$, $\alpha_a(I)$, respectively. Let x be the coordinates of a pixel on I, Then, its depth value is fused by

$$\alpha(x) = w_m \alpha_m(x) + w_d \alpha_d(x) + w_a \alpha_a(x), \qquad (3)$$

where w_m, w_d, w_h are the fusing weights for each cue. In this paper, $w_m = w_d = w_a = 1/3$. Fig.6(b) shows an example depth map by the multi-cue estimation.

Next we introduce depth estimation from each cue.

Depth from aerial perspective cue. As described in [8], the irradiance attenuates along the sight in a scene. Here we use it as one depth cue.

$$\alpha_h(x) = -\epsilon \ln t(x), \qquad (4)$$

where ϵ is the scattering coefficient of the atmosphere and $t(x)$ is the medium transmission, which is estimated using the dark-channel prior proposed in [8],

$$t(x) = 1 - \omega \min_c (\min_{y \in \Omega} \frac{I^c(y)}{A^c}), \qquad (5)$$

where ω is a constant parameter $0 < \omega < 1$, y is a pixel in a local patch centered at x, $I^c(y)$ is the intensity value in the color channel c (in RGB color space), and A^c is the atmospheric light intensity (please refer to [8] for the details of estimating A^c).

Pseudo-depth from motion. Here we make very simple assumptions that object with smaller motion is further away from the viewing point,

$$\alpha_m(x) = 1 - \frac{m(x)}{\max_{x \in I} m(x)}, \qquad (6)$$

where $m(x)$ is the optical flow magnitude at pixel x.

Although the assumption is simple, it is generally true especially for background, as the foreground objects are treated particularly in the system at a later stage (see Section 4.3).

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. In our system, we estimate depth from defocus as follow,

1. 2D wavelet transform is applied to a frame at three scales.

2. The degree factor is defined as

$$\varepsilon(x) = \frac{1}{|\partial x|} \sum_{y \in \partial x} \omega(y)/F(y), \tag{7}$$

where y is a pixel in the neighbor ∂x of x, $F(x)$ is the maximum spectral energies of the wavelets at the three scales. $\omega(y)$ is the weight on pixel y's focus degree. $\omega(y)$ is computed as

$$\omega(y) = e^{-\frac{||x-y||}{s(f(x),f(y))}} \tag{8}$$

where $||\cdot||$ is the Euclidean norm, $f(x)$ is color feature vector in RGB, $s(f(x), f(y))$ is the Cosine distance function between the two color vectors at pixel x and y.

3. Then, the depth estimated by defocus cue is computed as

$$\alpha_d(x) = \frac{\varepsilon(x)}{\max_{x \in I} \varepsilon(x)}. \tag{9}$$

4.3 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 provide seeds for depth propagation to the other frames in a video shot. Although the above multi-cue fusion method can provide a good depth estimation for background, the depth values of foreground pixels can be inaccurate if the foreground is in motion. Moreover, cognitive studies [14] suggested that human visual system is more sensitive to the quality of foreground objects. With such concern, we add an interactive step for foreground depth refinement. The refinement takes the following steps, (i) interactive segmentation of foreground objects, followed by (ii) foreground depth re-estimation and interactive adjustment.

4.3.1 Foreground segmentation

The interaction for foreground segmentation is shown in Fig.4 (b), where scribbles are drawn to indicate the foreground region (in red) or background region (in yellow). To segment an object, we modify the GrabCut method [23] by adding depth information to the model. In the original GrabCut method, the segmentation label $\ell_n \in \{0, 1\}$ for each pixel is obtained by iteratively minimizing an energy function

$$E(\ell, k, \theta, I) = U(\ell, k, \theta, I) + V(\ell, I), \tag{10}$$

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 intensity. (Please refer to [23] for the details of the GrabCut method.) In our system, besides color, the depth values $\alpha(I)$ obtained from the multi-cue depth estimation module (described in Section 4.2) are also considered in both the data term and smoothness term. Hence, we modify the original GrabCut data term as

$$U(\ell, k, \theta, I, \alpha(I)) = \sum_x D(\ell_x, k_x, \theta, I(x), \alpha(x)), \tag{11}$$

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

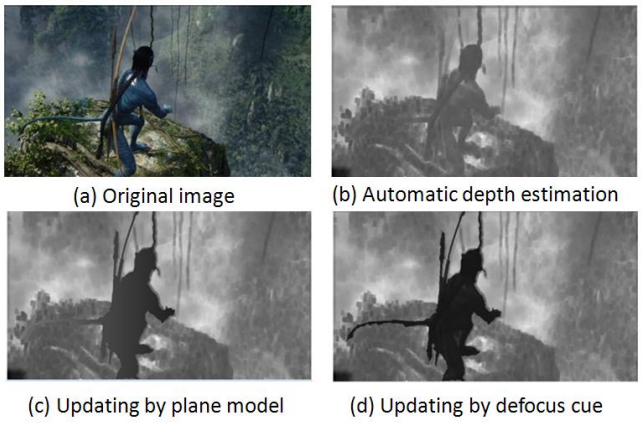

Figure 6: Depth estimation and foreground depth re-estimate.

The modified smoothness term is

$$V(\ell, I, \alpha(I)) = \sum_{x_m \in \partial x_n} S(x_m, x_n) \tag{12}$$

where $S(\cdot) = [\ell_m \neq \ell_n] e^{-\gamma_1 ||x_m - x_n|| - \gamma_2 |\alpha(x_m) - \alpha(x_n)|}$, and $[\ell_m \neq \ell_n]$ is 1 if $\ell_m \neq \ell_n$; otherwise 0. $||x_m - x_n||$ is Euclidean distance between two pixel coordinates. $|\alpha(x_m) - \alpha(x_n)|$ is the absolute difference of the depth, which penalizes neighboring pixels of different depth values.

If the GrabCut doesn't give an accurate segmentation, the system provide users with interactive interface to adjust object contours. We adopt Intelligent Scissors [19] to facilitate users to refine the segmentation contours efficiently. Figure 4 (c) shows the refinement of the hat contour by just dragging the three blue control points using Intelligent Scissors.

4.3.2 Foreground depth re-estimate

Foreground object depth and structure estimation can be inaccurate if the objects are in motion or no defocus cue can be leveraged. If the multi-cue depth estimation result is not satisfying for an foreground object, the system provides users to choose different methods to re-estimate the structure of an foreground object, i.e. (i) If the defocus cue is strong, user can select to use depth from defocus. An example is shown in Fig.6(d). (ii) If the object is static and the camera motion satisfies SfM constraint, user can use SfM. (iii) If no good cue to use at all, user can just adopt a plane surface model to represent the object structure as shown in Fig.6 (c).

Besides the structure of foreground objects, their locations in the scene may also need adjustment. Although the foreground depth by multi-cue fusion may be noisy, we assume that the majority of pixels are accurate; Hence, we can decide the object location/depth by majority voting the foreground pixels. Or user can manually adjust foreground object locations using the 3D labeling interface (shown in Figure 4 (a) window 4, and Figure 9).

4.4 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

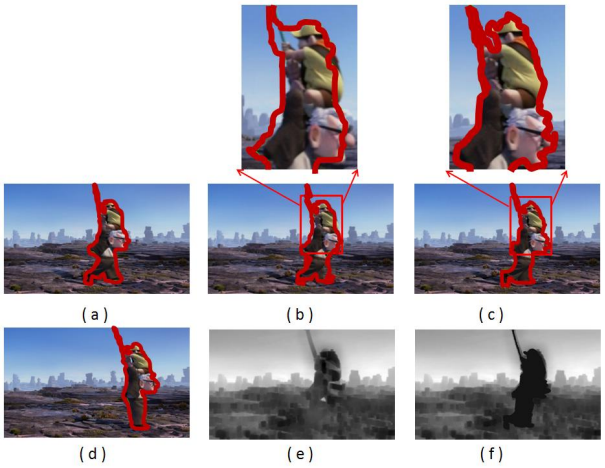

Figure 7: Foreground object tracking and depth update. (a) Labeled foreground object (red curve denotes its boundary) at a key frame $t-1$. (b) Warped foreground object contour from frame $t-1$ to frame t using KLT [18]. (c) Object boundary refinement by proposed level set method. (d) Tracked object at frame $t+1$. (e) Initial depth map estimated by multi-cue fusion at frame t. (f) The updated depth of foreground object after boundary refinement.

depth to the non-key-frames in the same shot based on object tracking and segmentation.

Given an foreground object and the depth map of a key frame, we first directly warp the object contours to the non-key-frames according to the object motion, and use the warped curve as initialization of the object location, and then evolve the warped contour to the object boundary at the non-key-frames using an adapted Level set method based on [3]. We introduce the details in the following.

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

$$\xi(\phi) = \mu R_p(\phi) + E_{img}(\phi) \tag{13}$$

where μ is a constant and the level set regularization term $R_p(\phi)$ can be written as:

$$R_p(\phi) = \int_\Omega p(|\nabla\phi|)dx, \tag{14}$$

where p is a potential function $p : [0, \infty) \rightarrow \Re$

$$p(\phi) = \frac{1}{2}\int_\Omega (|\nabla\phi| - 1)^2 dx. \tag{15}$$

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

$E_{img}(\phi)$ is the adapted term of the external energy in [3]. It depends upon the image data and depth map:

$$E_{img}(\phi) = aL_g(\phi) + bA_g(\phi) + c\Delta_\kappa(\phi) \tag{16}$$

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

$$L_g(\phi) = \int_\Omega g\delta(\phi)(|\nabla\phi|)dx \tag{17}$$

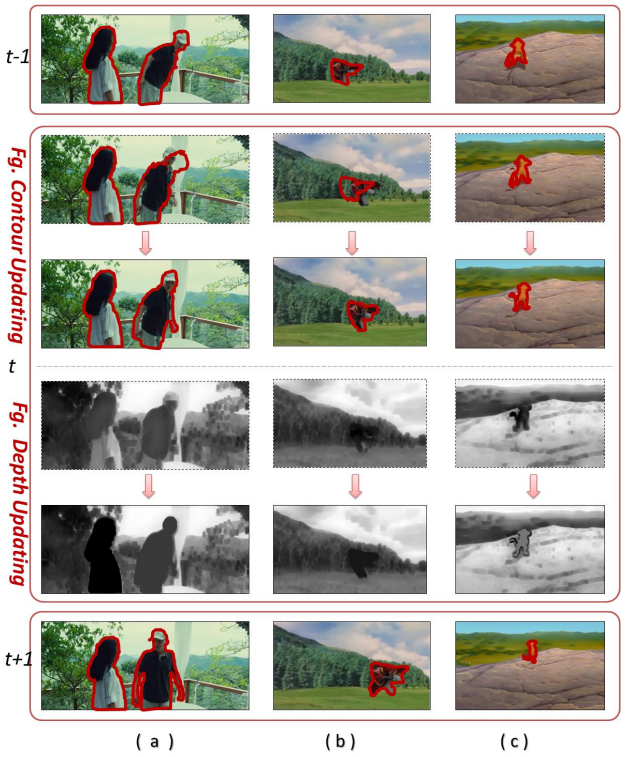

Figure 8: Results for depth propagation in a video shot.

$$A_g(\phi) = \int_\Omega gH(-\phi)dx \tag{18}$$

and

$$\Delta_\kappa(\phi) = \int_\Omega \kappa H(-\phi)dx \tag{19}$$

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

$$g = \frac{1}{1 + |\nabla G_\sigma * I|^2} \tag{20}$$

where I denotes for 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 map into the level set data term, the function κ is defined as

$$\kappa = \frac{1}{1 + \sum_{x,y\in N}|\alpha(x) - \alpha(y)|} \tag{21}$$

where N is a set of neighborhood pixels x and y in the domain Ω; $\alpha(x)$ is the depth value of pixel x.

The energy functional $\Delta_\kappa(\phi)$ is a new added term to the original formulation proposed in [3]. Assuming that pixels on an object share similar depths, it is proposed to estimate accurate contours of foreground objects according to the gradient field of a depth map in addition to the intensity map. $\Delta_\kappa(\phi)$ is introduced to penalize the case when the depth values inside the segmented object are quite different with each other. In experiments, it confirms that the final

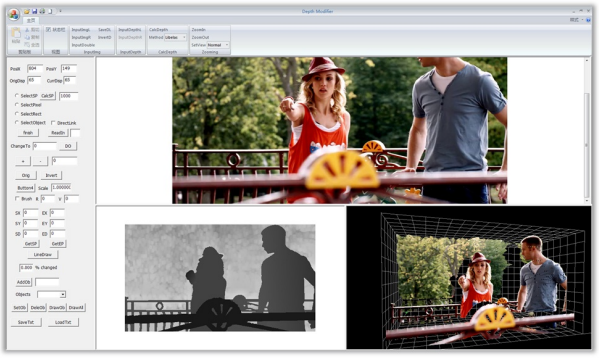

Figure 9: Depth labeling interface for evaluation.

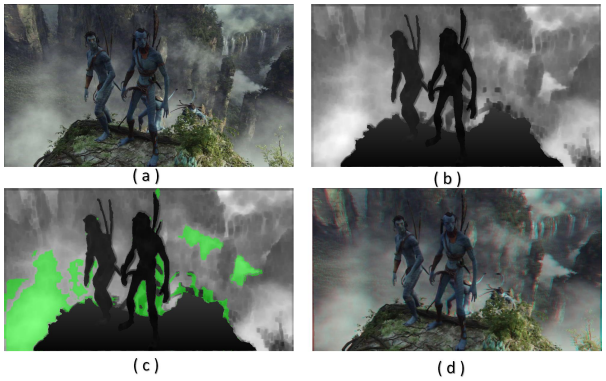

Figure 10: (a) Original frame (b) Depth map in our system (c) Difference between human labeled depth and the estimated one (d) the converted stereo frame

segmentation result is improved greatly by adding this term and the object's depth is more reliable after updated based on the refined segment, as shown in Fig. 7.

To speed up the evolution process and to obtain more refined result, we initialize the level set function based on the segmentation result of its previous frame. Given the segmented foreground object, we extract SURF(Speeded Up Robust Features) [2] as feature points for feature tracking and warp contour using KLT tracking [18] method, which is a fast, efficient, scale- and rotation-invariant interest point detector. The approximate 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 the general initialization. We apply a standard method to minimize the energy functional by finding the steady state of its gradient flow as [1].

More propagation results are shown in Figure 8. The first row shows the labeling result at frame $t-1$. The warping results using KLT [18] are shown in the second row. The direct warping seems inaccurate, especially when the displacement of the foreground objects is large, e.g. Figure 8(b). The third row displays the refined segmentation results by the proposed method. We can see the contours of the objects are localized very well. The fourth row shows the depth map estimated by the multi-cue fusion method described in section 4.2). In the fifth row, updated/re-estimated 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.

4.5 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 $d(x)$ at pixel x is the horizontal coordinate difference between the corresponding pixel in the left view and the right view of a stereo frame pair. When display, a pixel with negative disparity value is perceived as a point outside screen by viewers, and vice versa. A larger absolute value of $d(x)$ indicates a longer distance between the screen and the point.

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

$$d(x) = s \cdot W_I \cdot \left(\frac{\alpha(x) - \alpha_{min}(I)}{\alpha_{max}(I) - \alpha_{min}(I)} - \tau\right) \qquad (22)$$

where W_I is the image width, $\alpha_{max}(I)$ is the maxima of depth values. s is a control factor to restrict the maxima of absolute disparity, which 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%. $\tau (0 \leq \tau < 1)$ is a parameter that shifts the disparity to negative value and produce inside/outside screen effects. In our system, τ is determined by the stereo effect of a reference foreground object in the scene. Let x_{ref} be the reference point on a foreground object, and $\alpha(x_{ref})$ be its depth value. τ is computed as

$$\tau = \frac{\alpha(x_{ref}) - \alpha_{min}(I)}{\alpha_{max}(I) - \alpha_{min}(I)} - \frac{d(x_{ref})}{s \cdot W_I} \qquad (23)$$

$d(x_{ref})$ is the disparity of the reference point/object, which is automatically predicted by a trained disparity estimation model - a multi-label SVM .

We use motion and position of a segmented object as the features to predict its disparity value. The feature vector is composed of four components: (i) object motion magnitude and orientation histograms, (ii) pixel location histogram of the object region ,(iii) mean and variance of the depth values (by multi-cue estimation in Section 4.2) of the object points, and (iv) 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 3D movies, disparity maps are first directly computed by state-of-the-art stereo matching method [25]. Notice that we do not perform parallel view rectification before the stereo matching, hence, the disparity has signed value. Since the disparity values computed by stereo matching are always quantized into several disparity layers, we can use them as the labels in the multi-label Support Vector Machines (SVM). After extracted the motion and position features, the SVM is trained in one-vs-all manner. 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 2D video, then object disparity values are predicted.

During stereo view synthesis, an original 2D 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).

$$I_r(x) = I(x + 0.5 \times d(x)) \tag{24}$$

$$I_l(x) = I(x - 0.5 \times d(x)) \tag{25}$$

After warping, there appear some "holes" due to the discontinuity of the disparity values. An inpainting method [26] is utilized to fill these holes, and the system generates the final stereo views.

5. EVALUATION AND EXPERIMENTS

In experiments, we convert several well-known films into stereoscopic ones. Figure 11 shows the conversion results of key frames from six movie. Some propagation results are shown in Figure 7 and 8.

5.1 Evaluation method

The proposed depth estimation method is particularly suitable for professional movie, because in such high quality videos the three monocular cues are prevalent and can be fully exploited in estimating scene depth. Although he inexpensive cameras with depth sensors (such as Kinect) and the Make3D dataset [24] provide videos with corresponding depth maps, neither of them provides high quality videos. For this reason, we did not test the proposed depth estimation method using these videos and the corresponding depth maps as the ground truth data. Since there is lack of depth data from conventional movies, we propose to use human labeled depth maps as reference to assess the visual quality of converted stereoscopic video as follows.

1. Generating human labeled depth maps.
   Given an input video, we first use the proposed system to get the initial estimation of depth maps for every frame of the video, denoted as $\alpha_0(t)$. Then both the depth maps and the original frames $I(t)$ are given to a user. With the help of the interactive 3D labeling interface (Figure 9 shows the interface of the depth modifier.) and some guidelines, the user is asked to revise the depth maps according to his/her sense of depth and produce a refined depth map $\alpha_1(t)$, which is considered as the reference data.

2. *Objective evaluation* on depth maps.
   We use three objective criteria to evaluate the accuracy of estimated depth maps, i.e. the Mean Square Error (MSE), Peak Signal Noise Ratio (PSNR) and the percentage of pixels been modified.

$$MSE = \frac{1}{mnt} \sum_{t} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (\alpha_0(t,i,j) - \alpha_1(t,i,j))^2, \tag{26}$$

   where m and n are image height and width in pixels, respectively.

$$PSNR = 10 \log \frac{255^2}{MSE} \tag{27}$$

3. *Subjective evaluation* on stereoscopic videos.
   We adopt the Mean Opinion Score (MOS) in the Double Stimulus Impairment Scale (DISI) method to evaluate the visual effect of converted stereoscopic videos generated from the depth maps. Let the stereoscopic video produced by $\alpha_1(t)$ and $I(t)$ be the unimpaired reference and stereoscopic video produced by $\alpha_0(t)$ and $I(t)$ be the impaired one.

Table 1: Mean Opinion Score (MOS)

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

5.2 Evaluation setup and results

We use 6 clips from 6 different movies in evaluation. Figure 10 shows some of the key frames we use.

For the *objective evaluation* we use a 4% soft margin to simulate the depth of field (Dof) effect and the final results show that on average 24.5% pixels on the estimated depth maps are different from the reference data. And on average the MSE value between α_0 and α_1 is 1403.29 at 256 depth scales, and the average PSNR is 21.57dB. From figure 10 we can see that most of the difference takes place in textureless regions or on boundaries.

For the *subjective evaluation* we ask 13 subjects (trained experts) to report their subjective impression on the stereoscopic key frames shown to them. The subjects are presented with the stereoscopic key frame generated by system and the one by the human labeled reference depth map side by side. Then he/she is asked to vote on the first image using a 5-level impairment scale (from "impairments are imperceptible" to "impairments are very annoying"). Table 1 is a list of 5 level scores.

Our experiments report an average of 4.28 score of impairment. High score demonstrates the competence of our system of producing visually pleasing stereoscopic contents. Also notice that most of the time people could not perceive the difference in the background even some amount of pixels are different from the reference one. This implies that while watch stereoscopic view, people pay much more attention on the foreground objects and their depth order than that of the background, which attests our assumption that a better foreground will greatly improve visual impression. We also test these experts with a video shot in the same way and the average score increases to 4.58. This indicates that while watching image sequences, the artifacts can be even less noticeable.

5.3 Efficiency

The system is efficient in both interactive operations and the automatic modules. For a video with 1280×720 frames, the average time consumption of video is about $7s$ to $9s$ per frame, among which user interaction costs about $3 - 5s$ on a key-frame, and the automatic computation takes about 4s per frame. We list the details of the time consumption in Table2.

The IMAX system is a commercial system, its data are unavailable for us to compare. However, from media reports on the internet, the IMAX system takes about 6 to 10 weeks to convert a 2-hour 2D film into a stereo one; James Cameron used about 300 computer artists,60 weeks and 750,000 man

Table 2: Time consumption table

Item	Time(per frame)
Shot segmentation	$10ms$
Depth propagation	$3s$
Automatic depth estimation	$0.5s$
Depth to disparity conversion	$0.1s$
Foreground segmentation	$3s$ to $5s$
Total	$7s$ to $9s$

hours to convert the Titanic[10], while our system only takes about 10 days of PC-hours for automatic computation and 20-50 man-hours for user interaction.

Comparing to fully manual conversion systems, the proposed system is much more efficient due to the following reasons: (1) There are a number of automatic algorithms used in the system to reduce the labor of user interaction, e.g. the automatic background depth estimation, label propagation from key-frames to non-key frames, and disparity estimation; (2) The major user interaction is the foreground object segmentation on only key frames, but the segmentation algorithms are semi-automatic facilitated by a convenient U.I., i.e. the adapted Graph cut and intelligent scissors.

6. CONCLUSIONS

In this paper, we presented an interactive system of 2D-to-3D video converting. The system is comprehensive and it consists of a number of modules ranging from depth estimation, depth-to-disparity conversion, stereo view synthesis, video coding/decoding to audio-video splitting/merging. We also proposed a new evaluation method to assess visual quality of converted stereoscopic video. Experiment results demonstrate the advantage of the proposed system.

However, there are still some limitations in this method, e.g. the integrated monocular cues are still limited, they prefer high quality videos; and the fusion of depth estimation from different cues is simple (it is using 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 multi-cue fusion, so that it is able to robustly estimate scene depth from regular or even low quality videos as well. Then we can evaluate the performance of the proposed system using collected range data, i.e. by Kinect.

7. ACKNOWLEDGEMENT

We'd like to thank for the support from the following research grants 973-2009CB320904, NSFC-61121002 and NSFC-90920012. And thanks Yanhui Liang's help in part of the system implementation.

8. REFERENCES

[1] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. New York: Springer-Verlag, 2002.

[2] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *ECCV*, 2006.

[3] L. Chunming, X. Chenyang, G. Changfeng, and F. M. D. Distance regularized level set evolution and its application to image segmentation. *IEEE Transactions on Image Processing*, 2010.

[4] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *IJCV*, 2000.

[5] G. Guo, L. Liu, Z. Zhang, Y. Wang, and W. Gao. An interactive method for curve extraction. In *ICIP*, 2010.

[6] M. Guttmann, L. Wolf, and D. Cohen-Or. Semi-automatic stereo extraction from video footage. In *ICCV*, 2009.

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

[8] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. In *CVPR*, 2009.

[9] D. Hoiem, A. A. Efros, and M. Hebert. Closing the loop on scene interpretation. In *CVPR*, 2008.

[10] IEEE. Ieee and the titanic: A century of technological heritage and innovation, $http://www.ieee.org/about/news/2012/12april_2012.html$.

[11] IMAX. Imax 2d to 3d conversion, http://www.imax.com/corporate/technology/2d-to-3d-conversion/.

[12] D. Kim, D. Min, and K. Sohn. A stereoscopic video generation method using stereoscopic display characterization and motion analysis. *IEEE Trans. on Broadcasting*, 2008.

[13] 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 *ICIP*, 2007.

[14] J. J. Koenderink, A. J. van Doorn, A. M. Kappers, and J. T. Todd. Ambiguity and the 'mental eye' in pictorial relief. *Perception*, 2001.

[15] I. Koprinska and S. Carrato. Temporal video segmentation: A survey. *Signal Processing: Image Communication*, 2001.

[16] M. Liao, J. Gao, R. Yang, and M. Gong. Video stereolization: Combining motion analysis with user interaction. *IEEE. Trans. on Visualization and Computer Graphics*, 2011.

[17] C. Liu, H. Liu, S. Jiang, Q. Huang, Y. Zheng, and W. Zhang. Jdl at trecvid 2006 shot boundary detection. In *TRECVID*, 2006.

[18] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, 1981.

[19] E. Mortensen and W. Barrett. Intelligent scissors for image composition. In *Siggraph*, 1995.

[20] 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. on Circuits and Systems for Video Technology*, 2005.

[21] A. P. Pentland. A new sense for depth of field. *IEEE Trans. on PAMI*, 1987.

[22] E. Rotem, K. Wolowelsky, and D. Pelz. Automatic video to stereoscopic video conversion. In *SPIE*, 2005.

[23] C. Rother, V. Kolmogorov, and A. Blake. "grabcut.

[24] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3-D scene structure from a single still image. *IEEE Trans. on PAMI*, 2008.

[25] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithm. *IJCV*, 2002.

[26] A. Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics, GPU, and Game Tools*, 2006.

[27] J. Tighe and S. Lazebnik. Superparsing: Scalable nonparametric image parsing with superpixels. In *ECCV*, 2010.

[28] C. Tomasi. Shape and motion from image streams under orthography: a factorization method. *IJCV*, 1992.

[29] S. A. Valencia and R. M. Rodriguez-Dagnino. Synthesizing stereo 3d views from focus cues in monoscopic 2d images. In *Proc. SPIE*, 2003.

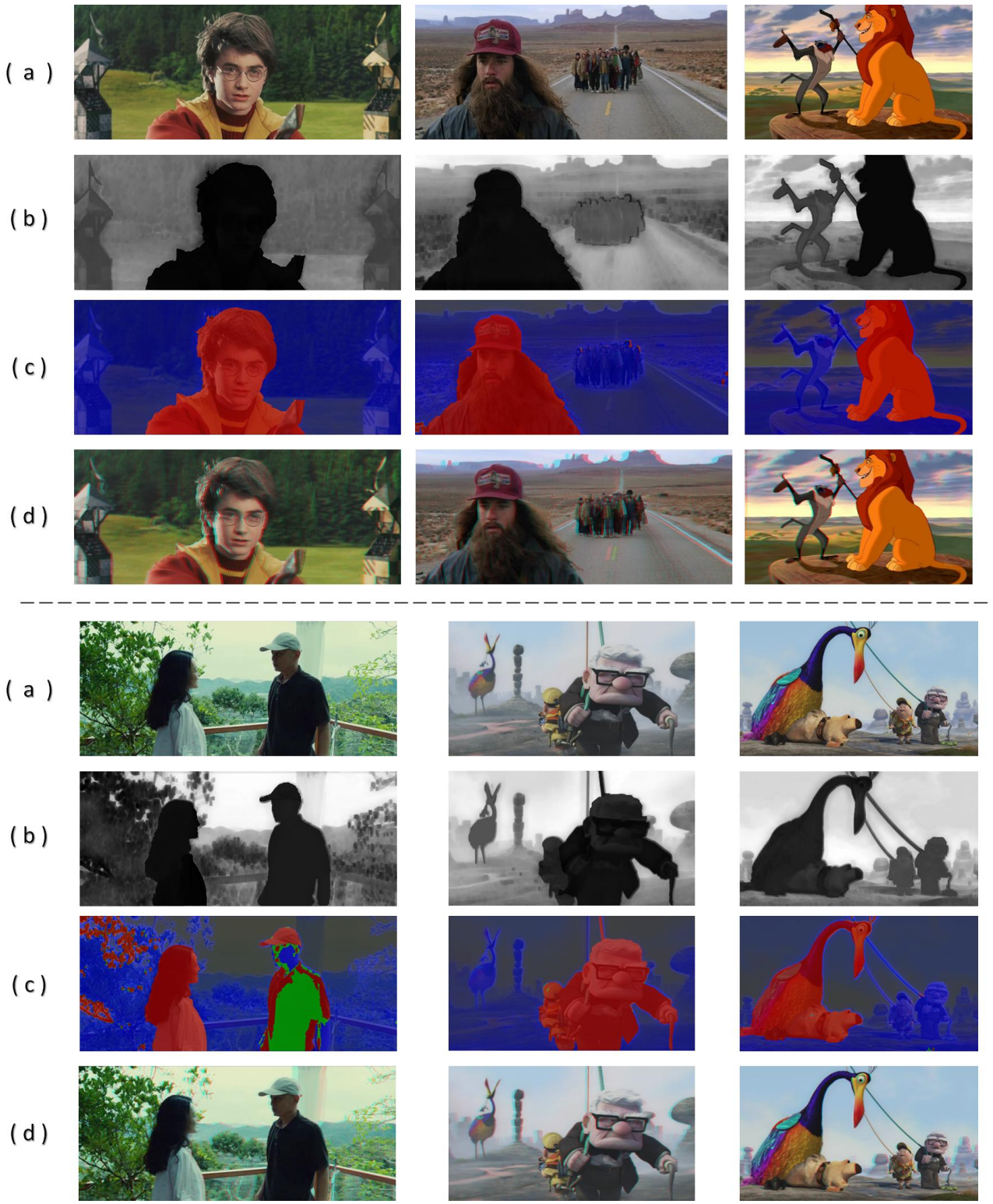

Figure 11: Some results. (a) Original frames. (b) System generated depth maps. (c) Stereoscopic effects illustration wrt virtual screen. Red, green and blue colors indicate outside, on and inside screen respectively. (d)Red-cyan anaglyphs.