A Compact Representation for Compressing Converted Stereo Videos

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

***Abstract*— We propose a novel representation for stereo videos namely 2D-plus-depth-cue. This representation is able to encode stereo videos compactly by leveraging the by-product of a stereo video conversion process. Specifically, the depth cues are derived from an interactive labeling process during 2D-to-stereo video conversion—they are contour points of image regions and their corresponding depth models, and so forth. Using such cues and the image features of 2D video frames, the scene depth can be reliably recovered. Experimental results demonstrate that the bit rate can be saved about 10%–50% in coding a stereo video compared with multiview video coding and the 2D-plus-depth methods. In addition, since the objects are segmented in the conversion process, it is convenient to adopt the region-of-interest (ROI) coding in the proposed stereo video coding system. Experimental results show that using ROI coding, the bit rate is reduced by 30%–40% or the video quality is increased by 1.5–4 dB with the fixed bit rate.**

***Index Terms*— Stereo video representation, stereo video coding, depth cue.**

I. Introduction

STEREO video technologies have been well acknowledged as the next milestone in digital video industry. Two major ones are stereo video generation and coding.

A. Stereo Video Generation

Usually, there are two approaches to acquiring stereo videos–one is directly capturing by stereo camcorders, the other is via 2D-to-stereo conversion. Stereo conversion is important because (i) the amount of directly captured stereo videos is not large enough to satisfy the demand of the stereo video industry, especially for 3DTV broadcasting; (ii) some classic conventional 2D movies or TV programs can be rejuvenated through the conversion.

In the market, the 2D-to-stereo conversion systems can be categorized into two classes, the automatic systems and the interactive ones. For example, real-time conversion chips developed by DDD have been built into the 3DTVs of Samsung, LG and Sharp, etc. However, the quality of converted stereo videos by such automatic systems is not always satisfactory. This is because that obtaining accurate depth is crucial to the conversion quality; whereas, depth estimation from a single view still remains an unsolved problem in computer vision. In order to generate high quality stereo videos it is necessary to introduce human in the loop. For instance, the IMAX's technology [1] is able to convert 2D live-action movies into stereo ones by using graphics models and human interventions. The technology has been successfully applied to converting IMAX Hollywood features, such as "Superman Returns" and "Harry Potter and the Order of the Phoenix".

In this paper, we study how to exploit the intermediate data derived from human interactions during the conversion so as to represent the converted stereo videos and improve the coding efficiency.

B. Stereo Video Coding

Compared with traditional 2D videos, stereo videos bring new challenges to the transmission and storage due to a larger data volume. There are two main categories of stereo video representations. One of them is based on the multi-view video (MVV) [12] representation, where the stereo video can be seen as a special case of MVV. The methods of multi-view video coding (MVC) encode stereo videos by exploiting the inter-view redundancy. The bit rate of the MVC is usually high due to the large amount of multi-view data. The other one is the 2D-plus-depth representation [26]. The coding methods based on 2D-plus-depth representation encode a 2D video and its frame-based depth maps with a relatively low bit rate. A 2D video and its depth maps are usually encoded independently in the conventional coding schemes, or only motion correlation between them is considered [28]. We believe that the bit rate of a stereo video can be further reduced by exploiting the correlation between the appearance and the structure of scenes. Besides, MVV and 2D-plus-depth representation both can be considered as special cases of another representation, called multi-view video plus depth (MVD) [21], which is used to represent the multi-view 3D videos.

Manuscript received March 28, 2013; revised December 29, 2013; accepted March 26, 2014. Date of publication April 4, 2014; date of current version April 22, 2014. This work was supported in part by the National Science Foundation of China under Grant 61231010, Grant 61272027, Grant 61300062, Grant 61121002, Grant 61210005, in part by the China Postdoctoral Science Foundation under Grant 2013M530482, in part by the International Postdoctoral Exchange Fellowship Program of the Office of China Postdoctoral Council under Grant 20130008 and in part by the Shenzhen Basic Research Program under Grant JC201104210117A and Grant JC201105170732A. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hassan Foroosh.

Z. Zhang, C. Zhou, Y. Wang, and W. Gao are with the National Engineering Laboratory for Video Technology, School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: zbzhang@pku.edu.cn; zhouch@pku.edu.cn; yizhou.wang@pku.edu.cn; wgao@pku.edu.cn).

R. Wang is with the Peking University Shenzhen Graduate School, Shenzhen 518055, China (e-mail: rgwang@szpku.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/TIP.2014.2315958

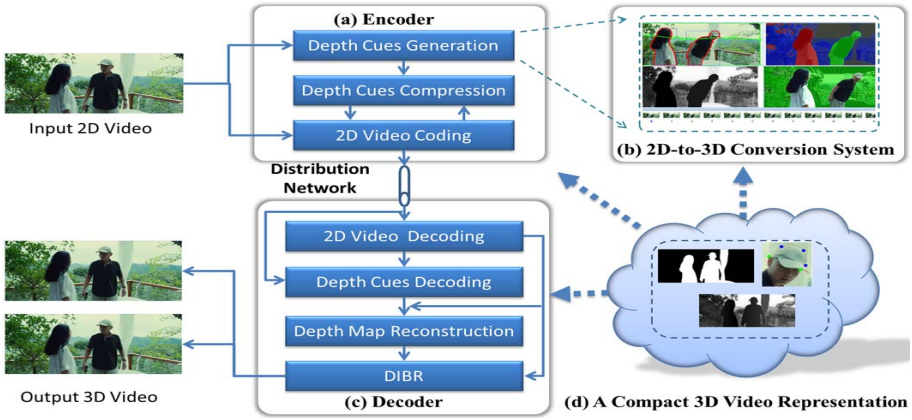


Fig. 1. The proposed system that coupling 2D-to-stereo video conversion and stereo video coding (CVCVC).

In this paper, we propose a novel stereo video representation to improve coding efficiency of stereo videos produced by 2D-to-stereo conversion. The representation consists of a *2D video* plus its *depth cues*. The depth cues are derived from the intermediate data during the operations of the 2D-to-stereo conversion process, including object/region contours and parameters of their designated depth model. Using the depth cues and by jointly considering the appearance of a 2D video, the depth of a scene can be reliably recovered. Consequently, two views of the derived stereo video can be synthesized. Compared with the depth maps, the depth cues are much more parsimonious than the frame-based depth maps. For example, compared with traditional stereo video representations, experimental results show that the bit rate can be saved about 10%–50% using the proposed representation.

To prove such an idea, we design a system and use the proposed representation to couple the stereo video conversion and video coding, namely *CVCVC*, as shown in Fig. 1. On the encoder side, depth cues are generated from the the "by-products" of an interactive conversion process [36] when converting a 2D video. Then, the 2D video and its depth cues are compressed jointly. On the decoder side, the 2D video and its depth cues are decoded from the bit-stream, then the depth cues are utilized to reconstruct depth maps according to image features of the 2D video. At last, the stereo video is synthesized using a DIBR method (e.g. [9], [34]).

In addition, since object contour is one of the components in the representation, it is convenient for the system to adopt the Region of Interest (ROI) coding to further improve the video quality given a limited bit rate, or to reduce the coding bit rate w.r.t. certain quality requirement. Experimental results show that compared with no-ROI coding, the bit rate is reduced by 30%–40%, or the video quality is increased by 1.5dB–4dB.

The remainder of this paper is organized as follows. In Sec. II, we survey the related work in both stereo video coding and 2D-to-stereo video conversion. In Sec. III, the new representation of stereo video is proposed. Based on the representation, we introduce the coding and decoding algorithms in Sec. IV. Experimental results and analysis are shown in Sec. V. Sec. VI concludes the paper.

II. Related Work

A. Stereo Video Coding

Stereo videos (two-view videos) can be seen as a special case of multiple-view videos. Hence, the multi-view video coding (MVC) methods can be directly applied to encode stereo videos. A key feature of the MVC scheme is to exploit both spatial and temporal redundancy for compression. A "P-frame" or "B-frame" can be either predicted from the frames in the same view using motion estimation (as the traditional video coding methods), or predicted from the frames of the other view so as to reduce the substantial inter-view redundancy [18]. MVC decoders require high-level syntax through the sequence parameter set (SPS in H.264/MPEG4 AVC) and related supplemental enhancement information (SEI) messages [5] so that the decoders can correctly decode the video according to the SPS. Compared with the simulcast coding scheme, the MVC generally achieves as high as 3 dB gain in coding (corresponding to a 50% bit rate reduction) [21]. For stereo videos, an average reduction of 20%–30% of the bit rate was reported [4].

The *2D-plus-depth coding* [26] is another type of stereo video coding. It is also called depth enhanced stereo video coding. The standard (MPEG-C Part 3) supports a receiver to reproduce stereo videos by depth-image based rendering (DIBR) of the second auxiliary video bitstream, which can be coded by the standard video coding scheme such as H.264/AVC [13]. The 2D-plus-depth is now supported by some stereo player softwares [8] and display devices, especially for the auto-stereoscopic displayer, e.g. Philips glass-free 3DTV and LG 3D mobile. Compared with the MVC solution, it is more convenient for 2D-plus-depth coding to render the stereo effects according to different devices.

Besides directly applying the standard encoder to coding the depth, researchers have also studied to exploit the characteristic of the depth map to compress it. An inter-component prediction method was proposed in [20]. The method derives block partition from the video reference frame corresponding to a depth map and utilizes the partition information to coding the depth map. Kim and Ho [14] proposed a new scheme to compress depth information of a mesh-based 3D video. Unlike previous mesh-based depth coding methods, they compress

TABLE I
EXAMPLES OF 2D-TO-STEREO VIDEO CONVERSION METHODS AND SYSTEMS

	System / Method	Representation (or Model)	Interactions (on key-frames)	Features	Intermediate By-products	Video Coding
Auto -matic Systems	"Visual Pertinent 2D-to-stereo Conversion" [34]	Pixels	N.A.	Motion, Aerial perspective	N.A.	N.A.
	"Stereoscopic Learning" [33]	Superpixels	N.A.	Motion, Location	N.A.	N.A.
	Tri-Def3D[8]	Pixel	N.A.	Color, Motion, Location	N.A.	2D+depth
Inter -active Systems	"An Efficient Conversion Method" [16]	Foreground objects, Background regions	Object segmentation depth assignment	N.A.	Object skeleton, Optical flow	N.A.
	"Semi-automatic Video Conversion" [31]	Foreground objects, Background regions	Object segmentation depth assignment	N.A.	Object contours, KLT features	N.A.
	"Stereo Extraction" [10]	Pixels	Depth layers labeling	Saliency, Motion	Depth classifiers	N.A.
	"Video Mesh" [3]	Triangular mesh of feature points	Alignment correction Occlusion labeling	Motion, Ground contact point	boundaries, Triangular mesh	N.A.
	"Interactive Stereo Video Conversion" [35]	Foreground objects, Background regions	Object segmentation Depth adjustment	Aerial perspective, Motion, Defocus	Depth cues, Object contours	N.A.
	"Video Stereolization"[17]	Moving objects, Background regions	Depth layers labeling	Motion	Moving Objects, Optical flow	N.A.
	"IMAX Conversion" [1]	3D object models	3D model pose & scale alignment	Information not released	Object & scene models	N.A.

the irregular depth information using a conventional 2D video coder, e.g. H.264/AVC. Cheung *et al* [6] present a unified optimization framework to combine two depth map compression techniques - graph based transform (GBT) and transform domain sparsification (TDS) for optimal coding performance.

In the literatures of 2D plus depth coding, there are two categories of the methods which are most related to our work.

One is extracting edges when coding the depth maps. Daribo *et al* [7] proposed a method to predict edge direction in a contiguous contour so that blocks along the contour can be divided into sub-blocks with different motions. Tabus *et al* [27] introduces an efficient method for lossless compression of depth map images, using the representation of a depth image in terms of three entities: 1) the crack-edges; 2) the constant depth regions enclosed by them; and 3) the depth value over each region. These methods are performed on the depth map and aims to efficiently encode depth video or reduce the artifacts around the edges. However, the edge information need to be transmitted to the decoder as a type of side information, which will degrade the coding performance, especially when the edge number increases. In this paper, we proposed an algorithm to compress the region contours.

The other one is using the correlation between depth maps and texture images to reduce the data redundancy. For instance, a codec proposed in [19] takes depth maps, images and edges as the input of the encoder. The codec encodes locations of depth edges explicitly and uses wavelet coefficients along depth edges to reduce the data entropy and to share edge information between the depth map and the image to reduce their correlation. Some methods directly down-sample depth maps to save the bit rate [23]. To solve the degradation problem caused by large scaling factors, [30] proposed a color based depth up-sampling method using the color frames as a prior to obtain high resolution depth maps on the decoder side. These methods have shown the correlation between depth maps and their corresponding texture frames can be utilized to improve depth maps compression. However, taking the color difference/similarity of neighboring pixels as prior to penalize their depth similarity/difference is not always useful, because a front-to surface (all pixels in the region share the same depth) could be full of textures and a textureless region could be slanted or curved.

In this paper, a higher level correlation [32] between depth maps and texture frames is utilized to further improve depth map compression in stereo video coding. A depth map is represented by a series of depth models covering different regions of an image. The region contours are further compressed into a set sparse control points. Then a depth cue is defined as a set of parameters derived from a certain depth model. Based on such definition, we propose a stereo video representation called 2D video plus depth cues.

B. 2D-To-Stereo Video Conversion Methods and Systems

The key of 2D-to-stereo video conversion is to obtain the depth map of each video frame so that stereo frame pairs can be synthesized. In a typical 2D-to-stereo conversion system, depth maps can either be estimated according to the appearance features of 2D frames, or be obtained with user assistance.

Four typical automatic conversion systems are listed in Table I. The *TransFantasy* system automatically converts 2D videos into 3D. In [34], motion and aerial perspective cues are leveraged to estimate depth maps. In [33], disparity maps, but not depth maps, are directly estimated by a trained SVM using features of object motion and context motion. DDD's *Tri-def3D* [8] converts videos in real time by estimating depth through analyzing features in a 2D video, e.g. the color, position and motion. The solution also supports different types of stereo video formats, such as side-by-side, bottom-up, and 2D-plus-depth, which can be encoded by any stereo video coding methods such as simulcast, MVC or 2D-plus-depth.

Several typical interactive systems are also listed in Table I. In such systems, usually, depth maps are interactively generated with user assistance. For instance, in the IMAX solution [1], graphics models are created for a scene by manually

aligning each object model to the corresponding pixels in a key frame, then the scene depth maps are generated according to camera poses. In order to reduce intensive user labeling, a semi-interactive conversion system [10] uses depth cues like visual saliency, scene motion and camera motion to recover scene depth. In [3], user assisted motion estimation and ground contact points are used to interactively reconstruct 2.5D triangular mesh of a scene. In [35], depth perception cues such as motion, aerial perspective and defocus are exploited to infer scene depth of each 2D video frame.

To alleviate the burden of storage and transmission, an efficient stereo video coding scheme is a necessary component in a 2D-to-stereo video conversion system. Since depth maps need to be generated during conversion, the 2D-plus-depth coding is widely adopted. However, in the current 2D-plus-depth coding scheme, a 2D video and its depth maps are encoded either independently, or only motion correlation between the two is considered [28].

In fact, other correlations can be further exploited to improve the compression ratio. For example, object boundaries can be used to confine both the appearance and the depth of the objects. In this paper, object boundaries obtained from the appearance features of 2D video frames are utilized as depth cues (to be introduced in Sec. III) to recover object depth in a scene. It is noted that object boundaries and the similar things are the by-products of the interactive conversion process. They can be very useful to ensure the conversion quality or improve coding efficiency. However, even with state-of-the-art computer vision algorithms, they usually cannot be reliably extracted from 2D videos automatically. In the 6^{th} column of Table I, there list a set of by-products from the operations/interactions (the 4^{th} column of Table I) in the typical interactive conversion systems.

Based on this consideration, we propose a novel representation of stereo videos in this paper. It enables the coupling of the two tasks, video coding and 2D-to-stereo video conversion, by utilizing the by-products of user interactions, so that the two tasks can inherently facilitate each other to improve stereo video compression efficiency.

III. The Proposed Stereo Video Representation

The motivation of of this paper is that the by-products (as shown in the 6^{th} column in Table I) generated in the stereo video conversion are with rich information to reconstruct the depth maps. For the stereo videos converted from 2D videos, some "depth cues" extracted from these by-products are sufficient to build the depths of the scenes. When storing and transmitting the converted stereo videos, we could use *2D videos plus depth cues* while not use *2D videos plus depth maps*. To prove this idea, we propose a system called CVCVC to couple the stereo conversion system and stereo video coding system together, as shown in Fig. 1. The conversion system is integrated in the encoder as the tool to generate the depth cues. A "depth map reconstruction" module is integrated in the decoder so that depth maps can be recovered from the depth cues.

Formally, the proposed stereo video representation is composed of a 2D video $V = (I_1, \ldots, I_t)$ of t frames, and a set of depth cues $\Gamma = (\Gamma_1, \ldots, \Gamma_t)$ for each frame. We name such a representation as *2D-plus-depth-cue*, denoted as

$$S = (V, \Gamma). \tag{1}$$

Depth cues Γ are used to implicitly describe the geometric scene structure of a 2D video frame. We assume scene structure is characterized by an *underlying* scene model $M(\Theta)$ with parameters $\Theta = (\theta_1, \ldots, \theta_t)$. In the encoding phase, a scene model M is appointed to V during the conversion operations ($\mathbb{O}$). According to the appointed model, the depth cues Γ are extracted from V and then transmitted with the compressed 2D video to the decoder. In the decoding phase, using the 2D video and its depth cues, the scene model parameter Θ of M is recovered, and then the depth maps D can be derived from M. Eq. 2 gives a summary of how the CVCVC system works, including model appointment by $\mathbb{O}$, depth cues extraction from videos given scene models, and the depth map recovering from depth cues Γ by referring 2D video frames. This representation can be seen as a variant of the *2D-plus-depth*. But in the 2D-plus-depth representation, V and the whole pixel-wise depth maps $D = (d_1, \ldots, d_t)$ are encoded and transmitted.

$$\underbrace{V \overset{\mathbb{O}}{\Longrightarrow} M(\Theta) \overset{V}{\Longrightarrow}}_{encoding\ phase} \quad \underbrace{\Gamma \overset{V}{\Longrightarrow} M(\Theta)}_{decoding\ phase} \Longrightarrow D \tag{2}$$

A scene model $M(\Theta)$ of a video is decided by the operations of the 2D-to-stereo conversion pipeline (as listed in Table I). In this paper, in order to prove the concept of the proposed representation, we adopt the system [36] and integrate it in the proposed CVCVC system as the conversion module,[1] as shown in Fig. 1(b).

For simplicity, the subscript of frame index t is omitted in the rest of the paper. Next, we introduce the scene model and how the model is appointed in the example conversion system in Sec. III-A, and then introduce the depth cues and how the cues are extracted in Sec. III-B.

A. The Scene Model $M(\Theta)$

In the conversion module, a video frame is segmented in to R regions, with region contours set $C = (C_1, \ldots, C_R)$. Each region corresponds to a semantic entity, e.g. an object, background, and we name such regions as *semantic regions*. Fig. 2 and Fig. 3 show the interfaces of labeling a foreground region and a background region, respectively. The foreground object is extracted using an interactive segmentation algorithm based on Graph-Cut method [2], [25].

Each semantic region has a model M_i for its depth, namely *region depth model*. Hence, the scene model of a frame is composed of a number of region depth models, i.e. $M = (M_1 \ldots, M_R)$. As listed in Table II, the region depth model is either a pixel-wise non-parametric model, (i.e. depth values in

[1] In fact, we can also use other interactive 2D-to-stereo video conversion systems as listed in Table I. The system [36] combines both interactive operations and automatic depth estimation in 2D-to-stereo conversion. We use this example to illustrate the proposed method could handle the both cases.

TABLE II
VARIOUS REGION DEPTH MODELS AND DEPTH CUES

	Model Flag	Model M_i	Model Parameters Θ	Depth Cues Γ	Instances	Operations $\mathbb{O}$
Non-parametric Model	$\mu_i = 1$	Pixel-wise depth	Region contours, Pixel depth	Region boundary control points, Monocular cue flags		Automatic estimation
	$\mu_i = 2$	Pixel-wise depth	Region contours, Pixel depth	Region boundary control points, Pixel depth		Interactive labeling
Parametric Graphics Model	$\mu_i = 3$	Planar surface	Surface contour, Depth on the contour	Surface contour control points, Surface normal direction, Depth of a reference point of the surface, Depth gradient of the surface	Walls, Ground	Interactively labeling
	$\mu_i = 4$	Polyhedron	Wire-frame	Occlusion boundary control points, End points of polyhedron surface boundaries	Vehicles, Buildings	Interactive model alignment and labeling
	$\mu_i = 5$	Face	3D graphics model	Object contour control points, Model pose and scale	Close-up Faces	Interactive labeling
	$\mu_i = 6$	Human body	3D graphics model	Object contour control points, Model pose and scale	Human actors	Interactive model alignment

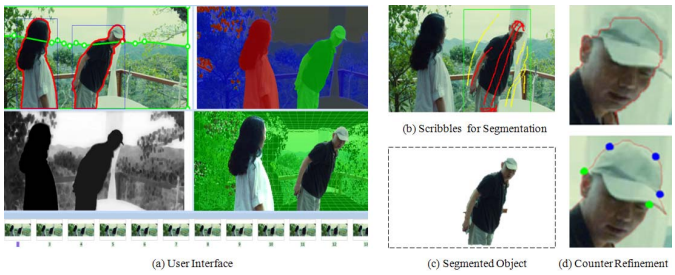


Fig. 2. A screen capture image of the user interface in [36] and its interactions for foreground objects segmentation.

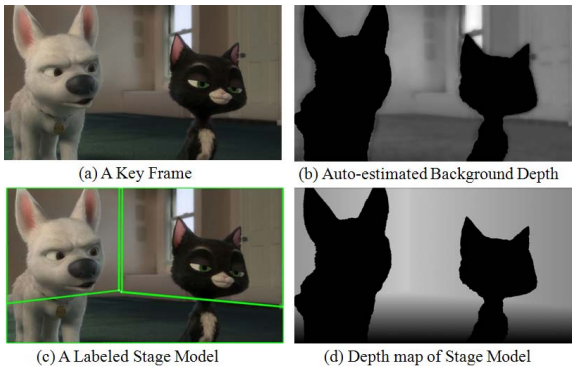


Fig. 3. Examples of two different background depth models of [36].

a region are either computed or manually labeled pixel-wisely,) or a parametric graphics model (such as planar surfaces, polyhedrons, stage models [29], and some specific models like human faces and human bodies [1], etc.). Such region depth models are appointed in the 2D-to-stereo video conversion system (listed in Table I), either interactively by the system users, or automatically by the estimation modules in the system.

The geometric configuration of each model M_i is characterized by a set of parameters Θ_i, as shown in Table II,

$$\Theta_i = (\mu_i, C_i, \psi_i), \tag{3}$$

where μ_i is the model type flag, C_i denotes the region contour of M_i, and ψ_i includes parameters used to compute the pixel depths in the region.

All the parameters of Θ_i are generated in the conversion module on the encoder side by the conversion operations $\mathbb{O}$. Here we introduce three typical region depth models in details.

1) Planar Surface Model ($\mu_i = 3$): Some objects can be modeled as a planar surface such as ground, walls, or even a human figure at certain distance. As shown in Fig. 2, the operations that create the model includes object segmentation to obtain the object/region contour C_i, surface normal adjustment, and surface depth assignment [36]. The depth value $\alpha(x, y)$ of a pixel on the surface is computed as

$$\alpha(x, y) = \alpha(x_g, y_g) + s_w\left(\frac{x - x_g}{w}\right)\sin\theta_v + s_h\left(\frac{y - y_g}{h}\right)\sin\theta_h, \tag{4}$$

where (θ_v, θ_h) is surface normal. (x_g, y_g) is the object's ground-standing point, i.e. the lowest point of object contour touching the ground. w and h are the width and height (in pixel) of an object bounding box (the green rectangle in Fig. 2(b)). The depth gradient is $(s_w \sin\theta_v, s_h \sin\theta_h)$, where (s_w, s_h) denotes the scale factor of depth gradient defined by users. Thus, the parameters of a planar surface model M_i are

$$\begin{aligned} \Theta_i &= (C_i, \mu_i = 3, \psi_i) \\ \psi_i &= (\alpha(x_{g_i}, y_{g_i}), x_{g_i}, y_{g_i}, w_i, h_i, s_{w_i}, s_{h_i}, \theta_{v_i}, \theta_{h_i}) \end{aligned} \tag{5}$$

2) Stage Model ($\mu_j = 4$): According to the research on visual perception [11], the partial order of distance is sufficient to describe the spatial relations among the objects far away, rather than using accurate depth values. Hence, in many real scenes, background regions are approximated by a generic model, namely stage model [29]. As shown in Fig. 3(c), a stage model consists of several planar surfaces, which is a special case of polyhedron. Each surface is defined by a set of end points of the intersection lines among the surfaces. The depth values of the end points are set according to user's perception of the scene geometry. Consequently, the parameters of a background stage model are

$$\begin{aligned} \Theta_j &= (C_j, \mu_j = 5, \psi_j) \\ \psi_j &= (p_{j_1} \ldots p_{j_P}, \alpha_{j_1} \ldots {}_{j_P}). \end{aligned} \tag{6}$$

C_j denotes the 2D region contour covering the whole stage on an image (e.g. the boundaries of the image). $(\alpha_{j_1} \dots \alpha_{j_P})$ are depth values of the end points $(p_{j_1} \dots p_{j_P})$. Depth values on each stage surface are pixel-wisely computed according to $(\alpha_{j_1} \dots \alpha_{j_P})$ through linear interpolation. Fig. 3(d) shows an example of recovered depth map using the stage model.

3) Non-Parametric Model ($\mu_k = 1$): The depth value of each pixel in a region can also be automatically estimated using computer vision algorithms, which exploit monocular depth perception cues extracted from the photometric features of images. The algorithm proposed in [36] is adopted in our system. The depth of a pixel p is computed by fusing three monocular photometric cues,

$$\alpha(p) = w_a \alpha_a(p) + w_m \alpha_m(p) + w_d \alpha_d(p), \tag{7}$$

where $\alpha_a(p)$ is the aerial perspective feature, $\alpha_m(p)$ is the motion feature and $\alpha_d(p)$ is the defocus feature. Users have the option to drop some of the cues if they observe that the cues are not reliable in certain images(Please refer to [36] for details of the algorithms). Then the model parameters are

$$\begin{aligned} \Theta_k &= (C_k, \mu_k = 1, \psi_k) \\ \psi_k &= (w_a, w_m, w_d). \end{aligned} \tag{8}$$

ψ_k indicates which cues are used and how the cues are combined to automatically estimate the depth in the region confined by C_k.

B. Depth Cue Extraction: Γ

Region depths can directly generated from the region depth models introduced in Section III-A. However, if there are many regions produced by the conversion operations or some regions are large, the data amount to represent the regions (points composing the region contour C_i) will increase, which will increase the coding length. It is necessary to further reduce points to represent the region contours.

So, the scene model M is an implicit representation of the depth of a scene. In the proposed representation, we use depth cues to represent the depth instead. Depth cues Γ_i are composed of two parts,

$$\Gamma_i = (CP_i, \mu_i, \psi_i). \tag{9}$$

(i) a set of sparse points CP_i, which is used to represent region R_i's contour C_i, and (ii) depth parameters (μ_i, ψ_i), which are used to compute the pixel depth values in R_i.

Here, because the number depth parameters is much smaller than the number of contour points, here (μ_i, ψ_i) are directly adopted from the model parameters (μ_i, ψ_i) as introduced in the previous section. However, compared with the point-wise representation of a contour, we use a set of sparse control points to represent the contour so that the data amount is reduced. In this paper, we design an algorithm (Alg. 1) to extract the control points from a region contour, called Lossless Contour Compression Algorithm.

Given such a representation, when converting 2D video to stereo one and coding the converted stereo video, a compact set of depth cues, Γ_i, is extracted from a video V for each semantic region R_i on the encoder side, subject to its designated region depth model M_i. On the decoder side, The region contour C_i can be recovered on decoder side from CP_i according to image gradients and Laplacians (we will introduce how the contour is recovered in Section IV-B).

Algorithm 1 Lossless Contour Compression

Require:
Image I;
A region contour $C = (p_1, \cdots, p_n, p_1)$
Ensure:
Contour control point set CP
Initialize $CP = (p_1)$, where p_1 is a random contour point;
$i = 1$;
while $i < n$ **do**
  for $j = i + 1$; $j \leq n$; $j++$ **do**
    Get a shortest path e_{ij} using the I.S. algorithm [22] on I between p_i and p_j
    if $e_{ij} = (p_i, \cdots, p_j)$ **then**
      continue;
    else
      add p_{j-1} into CP;
      $i = j - 1$;
      break;
    end if
  end for
end while

1) Control Points Extraction on Region Contours: Alg. 1 shows the proposed method to extract control points on a region contour by exploiting image gradient features. Here we use Intelligent Scissors (I.S.) [22] as a tool to "search" the control points (I.S. is originally an algorithm of interactive image segmentation). Alg. 1 extracts a minimum number of control points under condition that the region contour can be exactly recovered according to these points. Thus, the data amount of a region contour is reduced.

For a better understanding of Alg. 1, we first briefly introduce the I.S. method. An image lattice in I.S. is represented by a graph. The weight on a graph edge is defined as

$$f(p, q) = w_Z f_Z(q) + w_G f_G(q) + w_D f_D(p, q), \tag{10}$$

where w_Z, w_G and w_D are the weights that balance three terms f_Z, f_G, and f_D. The three terms are functions of image features—Laplacian zero-crossing, gradient magnitude of image intensity and gradient direction, respectively. A smaller edge weight indicates that a pair of neighboring pixels p and q tends to be on an object contour. Given any two nodes on the graph, p_{k_j} and $p_{k_{j+1}}$, a Dijkstra-like algorithm [7], [22] is used to search for the shortest path $\mathcal{P}^*_{k_j, k_{j+1}}$ between them by minimizing the cost

$$F(\mathcal{P}_{k_j, k_{j+1}}) = \sum_{(p,q) \in \mathcal{P}_{k_j, k_{j+1}}} f(p, q), \tag{11}$$

which is the sum of graph edge weight on each searching path.

Intuitively, if a contour fragment between two contour points can be recovered according to image gradients, then the fragment can be represented by the two points.

Given a region contour, I.S. iteratively searches for the longest contour section. It starts from a random contour point as the first control point, then sets the adjacent contour point as a candidate control point. Next, a contour section between the two points is recovered according to the image gradients between them using the I.S. algorithm. If the recovered contour section is exactly the same as the original contour, the candidate is not accepted as a control point; otherwise, its previous contour point is taken as a control point. The algorithm continues till all the contour points being traversed.

Alg. 1 can be considered as a *lossless* contour compression algorithm, since it guarantees that a region contour can be *exactly* recovered from the control points. We argue the following axiom and proposition are true:

Axiom 1: A sub-path of a shortest path is a shortest path.

Proposition 1: Under the condition of lossless compression, the number of control points selected by Alg. 1 is "nearly optimal", i.e. the number of selected control points is either minimum or at most exceeds the minimum number by 1.

Proof: Suppose $C = (p_1, \ldots, p_n, p_1)$ is a region contour with n pixels. $CP = (p_{k_1}, \ldots, p_{k_N}, p_{k_{N+1}})$ is a sequence of N control points extracted from C by Alg. 1, where k_i ($k_i \in \{1, \ldots, n\}$, $k_1 < k_2 < \ldots < k_N$) is a pixel index of a control point in C, and $k_{N+1} \equiv k_1 \equiv 1$. Assume the optimal control point sequence of C is $CP^* = (p_{k_1^*}, \ldots, p_{k_{N^*}^*}, p_{k_{N^*+1}^*})$, with the minimal control point number N^*. k_i^* ($k_1^* < k_2^* \ldots < k_{N^*}^*$) is also a pixel index on C.

A new sequence $CP' = (p_{k'_1} \ldots, p_{k'_{N'}}, p_{k'_{N'+1}})$ is constructed as following. If $p_{k_1^*} = p_{k_1}$, $CP' = CP^*$; Otherwise ($p_{k_1^*} \neq p_{k_1}$), p_{k_1} is inserted into CP^*, thus $N' = N^* + 1$, $k'_{N'+1} \equiv k'_1 \equiv k_1 (\equiv 1)$, $k'_i = k^*_{i-1}$ $\forall\ i \in \{2 \ldots N'\}$.

In order to prove the proposition, we show that $N \leq N' \leq N^* + 1$.

$N' \leq N^* + 1$ is obvious from the construction of CP'. Now we show $N \leq N'$ by showing that $k_i \geq k'_i$, i.e. each control point selected by Alg. 1 does not precede the corresponding point in CP'. This can be proved by induction:

(i) $p_{k'_1} = p_{k_1}$ holds according to the construction of CP'.

(ii) Denote the shortest path from point p to q as $\kappa(p, q)$, now we prove that if $k_{i-1} \geq k'_{i-1}$, then $k_i \geq k'_i$. This is proved by contradiction.

If $k_i < k'_i$, according to Alg. 1, the shortest path $\kappa(k_{i-1}, k_i + 1)$ is different from the original contour. However, note that the $\kappa(k'_{i-1}, k'_i)$ is the same as the original contour, and being its sub-path, $\kappa(k_{i-1}, k_i + 1)$ must be the same as the original contour as well (note that $k_{i-1} \geq k'_{i-1}$ is the induction premise, and $k_i + 1 \leq k'_i$ is the assumption), which results in a contradiction.

(iii) Step (ii) repeats till $i = N' + 1$. (i.e. $k_{N'+1} \geq k'_{N'+1}$.) And as $k'_{N'+1}$ reaches the end of control point sequence CP', $k_{N'+1} = k'_{N'+1} = k'_1 = k_1 (\equiv 1)$. So the number of points in CP cannot exceed that in CP', i.e. $N \leq N'$.

Proof Done.

2) Lossy Compression of Region Contours: The set of control points extracted by Alg. 1 is compact. For example, a region contour with about 2000 contour points can be represented by 200 $\sim$ 300 control points. In order to further promote the compression ratio, the constraint in line 6 of Alg. 1 can be relaxed to

$$\textbf{If}\ \ D(e_{ij}, (p_i, \ldots p_j)) \leq T_c\ \ \textbf{then} \tag{12}$$

where

$$D(P, Q) = \max\{\sup_{p \in P} \inf_{q \in Q} d(p, q), \sup_{q \in Q} \inf_{p \in P} d(p, q)\}. \tag{13}$$

$D(P, Q)$ is the Hausdorff Distance [24] between two sets of contour points P and Q. $d(\cdot, \cdot)$ computes the Euclidian distance between two points. T_c is a threshold. In this paper, with $T_c = 1$, a contour of 2000 points can be compressed to $30 \sim 40$ control points (shown as the red dots in Fig. 4).

Such a lossy contour compression method improves the contour compression ratio at the cost of the contour recovery accuracy.

Algorithm 2 Contour Reconstruction From Lossy Compression by *Simulated Annealing*

Require:
A decoded 2D frame I;
A region contour $C = (p_1, \cdots, p_N)$ recovered by I.S. method [22];
Ensure:
A smoothed region contour C'
1: Set initial temperature $T = T_0$, final temperature T_c, and a sampling counter $W = 0$;
2: **while** $T > T_c$ **do**
3: Randomly sample a p_{k_i} from C;
4: Compute $F'(\mathcal{P}_{k_i})$ according to Eq.15;
5: Set $F_{min} = MAX$ and $p_{min} = NULL$;
6: **for** each p_{n_i} in the $K_s \times K_s$ neighborhood $N_{b_{K_s}}$ of p_{k_i} **do**
7: Compute $F'(\mathcal{P}_{n_i})$
8: **if** $F_{min} < F'(\mathcal{P}_{n_i})$ **then**
9: $F_{min} = F'(\mathcal{P}_{n_i})$;
10: $p_{min} = p_{n_i}$;
11: **end if**
12: **end for**
13: **if** $F_{min} < F'(\mathcal{P}_{k_i})$ **then**
14: $p_{k_i} = p_{min}$
15: **else**
16: Randomly sample new p'_{k_i} in $N_{b_{K_s}}$;
17: **end if**
18: Compute $F'(\mathcal{P}_{k'_i})$;
19: **if** $\exp\{\frac{F'(\mathcal{P}_{k'_i}) - F'(\mathcal{P}_{k_i})}{T}\} > Rand[0, 1)$ **then**
20: $p_{k_i} = p_{min}$
21: $W = W + 1$;
22: **end if**
23: **if** $W > \varepsilon \times K_s^2$ **then**
24: break;
25: **else**
26: $T = 0.8 \times T$
27: **end if**
28: **end while**

(a) Directly recovered by IS

(b) Reconstructed Contour by our algorithm

Fig. 4. Contour reconstruction from the control points extracted by the lossy contour compression method. (a) The contour reconstructed by I.S. (b) Smoothed contour using the proposed Alg. 2.

IV. Stereo Video Coding and Decoding

In this section, we introduce how to use the *2D-plus-depth-cue* representation in a stereo video coding/decoding system (as shown in Fig. 1).

A. Encoding

1) Encoding of a 2D Video: In our system, a 2D video is encoded using H.264/AVC High profile (any other standard 2D video coding method also can be adopted). It is noted that the encoding of a 2D video, particularly the lossy encoding, may affect the reconstruction of depth cues. Thus, a new rate distortion cost is defined by considering distortion of recovered region contours,

$$J = D_{2D} + D_{cue} + \lambda B, \tag{14}$$

(a) The proposed method (b) 2D-plus-depth

Fig. 5. Examples of depth maps of our method and the 2D-plus-depth with QP value of 40.

where D_{2D}, B and λ denote the decoded 2D video distortion, encoded bit count and a Lagrangian multiplier, respectively. D_{cue} is the distortion of recovered region contours measured by the Hausdorff Distance (Eq. 13) between the original contour and the recovered one.

2) Encoding the Depth Cues $\Gamma = (CP, \mu, \psi)$: The depth cues are considered as the "user-custom SEI" in H.264/AVC bit-stream and encoded by fix-length coding. For example, the coding length of a stage model is usually within $2Kb$; For a non-parametric model, only the model type flag $\mu_k = 1$ and the control points set CP_k are encoded.

There is a strong correlation between corresponding region contours of consecutive frames. However, because the region contours are represented by control points, encoding the points takes about the same bits as encoding their motion vectors. The gain by considering the temporal correlation of the depth cues is trivial, especially for deformable or articulated objects/regions. Hence, the depth cues are encoded frame by frame independently in our system.

B. Decoding

As shown in Fig. 1, on the decoder side a stereoscopic video is synthesized and then displayed. It takes the following steps: (i) A 2D video is first decoded from the bit-stream by a H.264/AVC decoding module, and its depth cues are extracted from the "user-custom SEI". (ii) For each frame, region contours and the depth model parameters for each region are recovered according the depth cues and the video. (iii) The depth map is reconstructed by computing the pixel depth values according to the model and its parameters of each region. (iv) Finally, stereoscopic frames are synthesized using a DIBR method (e.g. [9]).

For example, for a non-parametric scene model with automatic depth estimation ($\mu_k = 1$), the depth value of each pixel is recovered by the corresponding automatic depth estimation algorithm used on the encoder side (Eq. 7). If a region is modeled as a planar surface ($\mu_k = 3$), the depths of the region are recovered using the model parameters in Eq. 4.

Region contours can be recovered section by section using Intelligent Scissors between each pair of neighboring control points by referring to the image gradients of the decoded frames. However, when Alg. 1 is relaxed to lossy compression (as described in Sec. III-B.2), the recovered contours are prone to *zigzag artifacts*, especially when the image gradients around the contours are weak (as shown in Fig. 4(a)). Thus, Alg. 2 is proposed to alleviate this artifact (as shown in Fig. 4(b)). It enforces a smoothness constraint between two adjacent curve fragments on the contour. Compared with the contour path energy defined in Eq. 11, here the energy of connecting two curve fragments, $(p_{k_j-1}, \ldots, p_{k_{j+1}})$ and

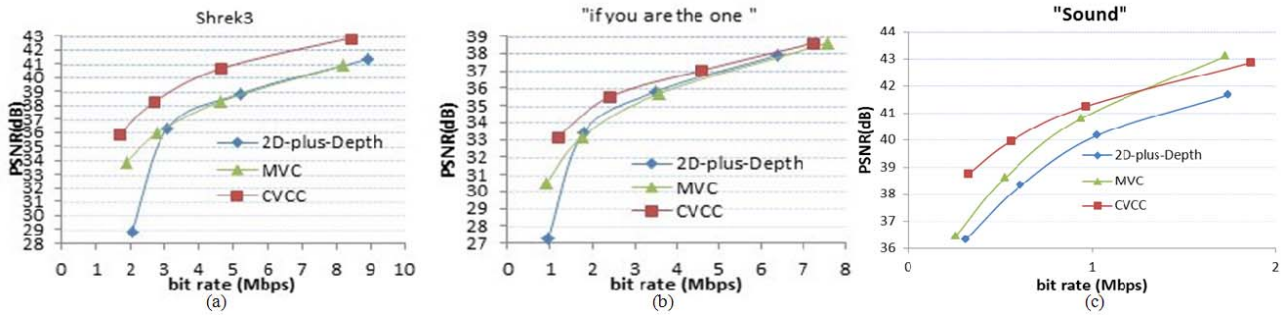


Fig. 6. Rate distortion comparison among the three stereo video coding methods.

TABLE III

CODING EFFICIENCY COMPARISON AMONG CVCVC, 2D-PLUS-DEPTH AND MVC

Test sequences	Resolution	QP	MVC		2D+ Depth			CVCVC		
			Total bit rate (kbps)	Average distortion (dB)	Depth maps bit rate (kbps)	Total bit rate (kbps)	Average distortion (dB)	Depth cues bit rate (kbps)	Total bit rate (kbps)	Average distortion (dB)
"Shrek3"	1080p	28	8190.29	40.90	4872.13	8889.66	41.34	120.33	8392.93	42.85
		32	4640.23	38.30	2933.21	5190.53	38.84	118.79	4635.82	40.67
		36	2783.38	35.95	1783.63	3094.64	36.34	110.21	2707.14	38.25
		40	1875.81	33.86	1191.37	2046.97	28.82	120.89	1645.82	35.85
"if you are the one"	1080p	28	7560.23	38.60	1377.82	6389.06	37.91	60.56	7240.78	38.59
		32	3575.14	35.68	801.33	3495.97	35.79	59.23	4598.36	37.03
		36	1731.74	33.10	512.64	1774.36	33.40	62.21	2394.65	35.48
		40	907.77	30.53	293.34	949.45	27.27	61.82	1168.94	33.17
"Sound of Music"	720p	28	1724.54	43.08	837.23	1739.87	41.66	57.02	1867.44	42.85
		32	936.83	40.78	482.61	1022.59	40.14	55.11	967.13	41.25
		36	524.50	38.59	297.04	610.83	38.31	52.01	558.31	39.95
		40	257.50	36.43	151.17	311.83	36.31	50.23	326.53	38.75
"City Angels"	720p	28	1582.82	43.10	765.80	1665.48	42.55	23.40	1450.42	43.98
		32	938.17	40.63	552.10	1090.64	40.39	23.24	900.72	41.47
		36	590.94	38.21	411.59	751.05	37.97	25.37	682.13	39.02
		40	386.26	35.67	304.74	616.57	36.31	28.22	520.53	37.88
"Lara"	720p	28	2500.65	45.07	619.68	1883.48	44.41	59.08	1792.44	44.95
		32	1652.68	42.98	454.90	1287.43	42.62	59.33	1234.15	43.72
		36	1152.63	40.66	346.07	930.81	40.50	59.87	940.31	42.03
		40	841.31	38.16	265.93	691.34	38.14	58.35	713.22	39.75
"This Is It"	1080p	28	10320.94	43.39	592.62	6163.62	42.65	35.63	7284.44	43.73
		32	5586.59	40.49	460.38	3691.16	39.87	36.42	3807.11	42.51
		36	3131.16	37.85	382.42	2249.96	37.15	38.50	2316.41	40.63
		40	1791.23	35.12	323.37	1399.99	34.47	40.46	1431.53	37.82
"Wall-E"	1080p	28	5608.66	39.42	1436.68	4852.48	38.90	0.24	4987.33	41.62
		32	2823.80	36.94	967.06	2738.52	36.32	0.24	2698.27	39.01
		36	1618.29	34.62	695.30	1723.92	33.81	0.24	1709.12	36.93
		40	1030.36	32.26	506.03	1162.86	31.32	0.24	1073.53	34.25

$(p_{k_{j+1}}, \ldots, p_{k_j})$, is defined as

$$F'(\mathcal{P}_{k_{j-1},k_j,k_{j+1}}) = F_{IS}(\mathcal{P}_{k_j,k_{j+1}}) + F_{IS}(\mathcal{P}_{k_{j-1},k_j}) + \gamma\, F_{smooth}(\mathcal{P}_{k_{j-1},k_j,k_{j+1}}), \quad (15)$$

where F_{IS} (defined in Eq. 11)) is the reconstruction energy of a contour section between two points using I.S. The smooth term F_{smooth} penalizes the direction difference between two adjacent contour curves,

$$F_{smooth} = \frac{1}{2}\sum_{i=0}^{M_k}\left(1 - \frac{\mathbf{r}_i \cdot \mathbf{r}_{i+1}}{\|\mathbf{r}_i\|\|\mathbf{r}_{i+1}\|}\right), \quad (16)$$

where $\mathbf{r}_i$ denotes the direction of a curve fragment,

$$\mathbf{r}_i = \begin{cases} (p_{i+K_s}, p_i), & i + K_s \le N; \\ (p_N, p_i), & i + K_s > N. \end{cases} \quad (17)$$

γ balances F_{smooth} and F_{IS}.

TABLE IV

CODING EFFICIENCY IMPROVEMENT BY CVCVC

Test sequences	CVCVC vs. MVC		CVCVC vs. 2D+ Depth	
	Bit rate saved	Quality improved	Bit rate saved	Quality improved
"Shrek3"	55.23%	3.43 dB	55.48%	3.99 dB
"if you are the one"	14.24%	0.99 dB	19.25%	1.19 dB
"Sound of music"	29.70%	0.77 dB	48.87%	1.61 dB
"City Angels"	19.34%	1.09 dB	33.38%	2.22 dB
"Lara"	32.82%	2.08 dB	19.71%	1.26 dB
"This is it"	58.92%	3.24 dB	50.69%	2.48 dB
"Wall-E"	38.72%	2.21 dB	43.33%	2.94 dB

The local points around the contour are sampled by a Simulated Annealing [15], as shown in Alg. 2.

V. EXPERIMENTS AND RESULTS

A. Experiment Settings

Seven 2D movies are used as the testing data, including two $1080p$ cartoon movies, two $1080p$ regular movies, and three

Fig. 7. Comparison between the Frames of ROI and Non-ROI Coding of three sequences (a) (d) (g), (b) (c) (e) (f) (h) (i) are enlarged regions for comparing in details.

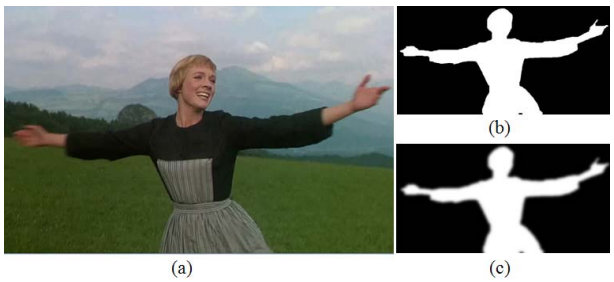


Fig. 8. CVCVC for ROI coding. (a) A video frame. (b) Labeled foreground object (ROI) mask. (c) Blurred ROI mask. If the ROI mask is not blurred, there will be a sharp boundary between the ROI and region of non-interest.

$720p$ regular movies. Each video has 90 frames with a frame rate of $30fps$ and GOP structure of "IPPP...". Five reference frames are used and the search range is set to 32. Experiments are performed on computers with 4G RAM and Intel Core 2 Duo CPU E8400 3.00GHZ.

In the proposed CVCVC system, H.264/AVC High profile is adopted to encode the 2D videos. The depth cues are encoded as the "user-custom SEI" in H.264/AVC bit-stream.

The CVCVC is compared with the other two stereo video coding methods, the MVC and 2D-plus-depth. To evaluate the performance, depth maps and stereo videos are generated before the encoding according to the extracted depth cues. For 2D-plus-depth, the original 2D videos and the depth videos are encoded by H.264/AVC High Profile. On the decoder side, stereo videos are synthesized to evaluate the distortion. For MVC, two views of the synthesized stereo videos are encoded

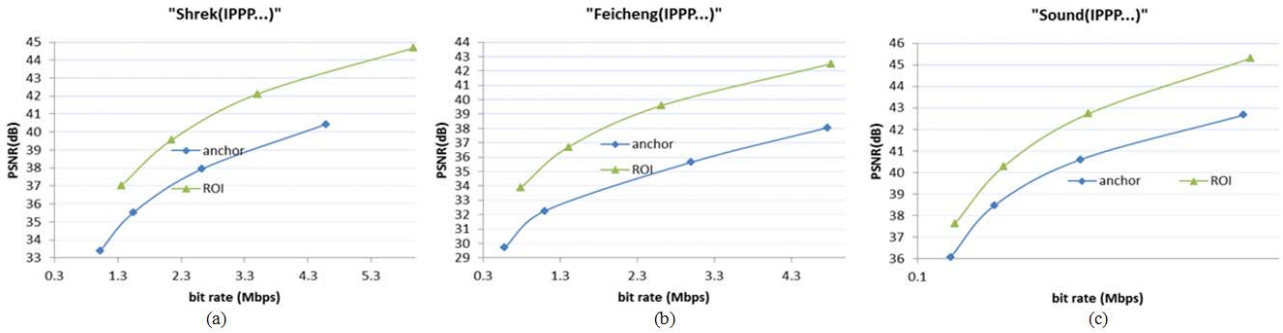


Fig. 9. Rate distortion comparison between ROI and Non-ROI coding of three sequences (a) (b) (c).

TABLE V
COMPARISON BETWEEN ROI CODING VS NON-ROI CODING METHODS

Test sequences	QP	Non-ROI		ROI	
		Total bitrate (kbps)	Average distortion (dB)	Total bitrate (kbps)	Average distortion (dB)
"Shrek3"	24	4589.26	40.43	5971.88	44.68
	28	2629.68	37.94	3508.92	42.11
	32	1551.90	35.53	2145.90	39.54
	36	1023.48	33.40	1358.78	37.02
"if you are the one"	24	4772.62	38.05	4813.94	42.48
	28	2996.54	35.64	2616.30	39.58
	32	1099.00	32.25	1407.23	36.70
	36	579.76	29.70	789.20	33.89
"Sound of music"	24	912.02	42.66	929.88	45.29
	28	506.3	40.60	526.15	42.72
	32	290.98	38.46	313.90	40.27
	36	182.54	36.08	193.70	37.64

TABLE VI
ALGORITHM COMPLEXITY IN CVCVC

	Algorithms	Complexity
Encoder	Intelligent Scissors	$O(\|E\|)$
	Control Points Extraction - Alg. 1	$O(n^3)$
Decoder	Contour Reconstruction by I.S.	$O(n^2)$
	Contour Reconstruction by Alg.2	N.A.

by "Stereo High Profile" implemented in the latest H.264/AVC reference software - JM18.0. The average distortions between the stereo videos at the decoder and the synthesized ones at encoder are computed.

B. Results and Comparison

1) Depth Map Generation: As described in Sec. III-A, different region depth models can be applied to the scenes. Fig. 3 shows the depth maps of the same scenes recovered by the stage model and the non-parametric model using automatic depth estimation. Fig. 5(a) displays another scene, whose complex background depth model is estimated by an automatic algorithm [34] and the foreground object is represented with a simple planar surface model. These results show the flexibility of the proposed CVCVC system in estimating scene depth.

2) Coding Performance Comparison: QP values of 28, 32, 36 and 40 are used for the MVC and 2D-plus-depth to generate the coded videos and depth videos with different bit rates. They are also used to encode the 2D videos in the CVCVC. Since the bit rates of depth cues in the CVCVC are within 150 kbps, the saved bits are reallocated to 2D videos.

The rate distortions of the three methods are shown in Table IV.[2] Compared with the MVC, the CVCVC improves the decoded stereo video quality, on average, by 0.77dB–3.43dB at the same bit rates. Or it saves 14.24%–58.92% bit rate under the same decoded quality. Compared with 2D-plus-depth, the CVCVC improves the decoded stereo video quality by 1.19dB–3.99dB. Or it saves 19.25%–55.48% bit rate. Fig. 6 shows the performance comparison of three testing sequences. Details of the comparison are shown in Table III.

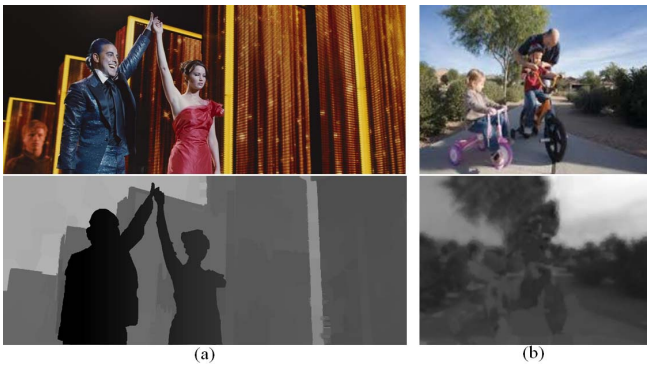
It is worth noting that sometimes the bit rate of coding a stereo video by the CVCVC can be dramatically reduced, especially when the geometric structure of the scene is more complex than the textures in the 2D video, e.g. cartoon videos. For example, the reduction of the coding bit rate of "Shrek3" by the CVCVC is more than 50% compared with the MVC and 2D-plus-depth. This reduction of the bit rate is due to using the parametric models or the automatic algorithms to estimate the depth maps. Examples of depth maps of our method and the 2D-plus-depth are shown in Fig. 5

C. Application: Region-of-Interest (ROI) Coding

The CVCVC inherently supports the ROI coding scheme, because there is an object segmentation step (shown in Fig. 2) in the conversion module of CVCVC. Fig. 8 show the process of ROI extraction. Hence, the video quality at ROI can be improved at low bit rates, or the coding bit rate can be further reduced while maintaining the ROI quality.

We compare the rate distortion of ROI coding with the anchor method (coding in CVCVC without ROI). Three sequences (each contains 90 frames) are tested in the experiment using four QPs of 24, 28, 32, and 36. As shown in Fig. 9, the ROI coding improves the decoded stereo video quality by

[2]For the sequence "Wall-E", in CVCVC, each frame is considered as a whole region without segmentation. The depth map of the frame is estimated by the non-parametric model in Sec. III-A.3. The depth cues consist of μ_k and ψ_k in Eq. 8. Thus, the bit rate of the depth cues is very low (0.24kbps).



(a) (b)

Fig. 10. Limitations of our method. (a) A complex scene where all regions in the frame are with the planar surface model. (b) All pixel depths are modeled as the non-parametric model.

2.59dB, 3.78dB and 1.78dB at the similar bit rate, or it saves 40.27%, 46.56% and 34.72% bit rates with the same quality on average. (Details are shown in Table V). Examples of decoded frames of ROI coding and non-ROI coding are shown in Fig. 7. The ROI coding improves the quality in ROI regions, as shown in the enlarged regions in 2^{nd} and 3^{rd} columns.

D. System Time Complexity

Table VI shows the time complexities of the algorithms used in the encoder and the decoder of CVCVC. On the encoder side, the time complexity of I.S. used in Alg. 1 is $O(|E|)$ [22], where $|E|$ is the edge number of the graph constructed in I.S. For the control point extraction algorithm (Alg. 1), the upper bound of complexity is $O(n^3)$, where n is the number of pixels on the generated region contours by the conversion operations. On the decoder side, the time complexity of contour reconstruction using I.S. is $O(n^2)$. The complexities of the two algorithms are only decided by pixels numbers on contours, no matter how many objects generated on one frame. For example, on a 1280×720 frame, it takes about 5.1s to extract 24 control points from a region contour about 2657 pixels on average. The computation time of recovering a region contour with 2657 pixels is $40 \sim 100$ms. For the Alg. 2, although the Simulated Annealing used in the algorithm is hard to analyze due to its stochasticity, it takes about 1s on average to reconstruct the region contour.

E. Discussion

One limitation of the proposed method might be concerned is the algorithm complexities on the decoder side, which cannot be performed in real time currently. However, since contour sections are independently reconstructed, the algorithms can be highly parallelizable. In our future work, we will study on speeding up the decoding by utilizing parallel-computing both on CPU and GPU.

Besides, As a system coupling coding and conversion together, the coding performance might be affected by the selection of the 2D-to-stereo conversion module and the interactive conversion operations. For example, in Fig. 10(a) some regions in the background of the scene are labeled even if there are no strong region contours, which undermines the performance of compressing region contour. In Fig. 10(b), the depth map are automatically estimated in [36] and modeled as non-parametric model in this paper. Although the depth can be transmitted using very low bit rate (below $1k$), the stereo effect will be degraded due to the fail of depth estimation.

VI. Conclusion

We proposed a novel compact stereo video representation, *2D-plus-depth-cue*. A system called CVCVC is designed for both stereo video generation and coding. Using the representation, the coding efficiency of the converted stereo video can be largely improved. This representation can also be extended and applied to other related fields, such as object-based video coding, ROI-based coding, and intelligent video content understanding. We showed several cases of applying the proposed representation to the ROI coding. The limitation of the propose methods and system are also discussed and we will study to improve them in our future work.

References

[1] IMAX. (2012). *2D to 3D Conversion* [Online]. Available: http://www.imax.com/corporate/technology/2d-to-3d-conversion/

[2] Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images," in *Proc. ICCV*, 2001.

[3] J. Chen, S. Paris, J. Wang, W. Matusik, M. Cohen, and F. Durand, "The video mesh: A data structure for image-based three-dimensional video editing," in *Proc. IEEE ICCP*, Apr. 2011, pp. 1–8.

[4] T. Chen, Y. Kashiwagi, C. S. Lim, and T. Nishi, *Coding Performance of Stereo High Profile for Movie Sequences*, document JVTAE022, Joint Video Team, London, U.K., 2009.

[5] Y. Chen, Y.-K. Wang, K. Ugur, M. M. Hannuksela, J. Lainema, and M. Gabbouj, "The emerging MVC standard for 3D video services," *EURASIP J. Adv. Signal Process.*, vol. 2009, p. 786015, Mar. 2009.

[6] G. Cheung, W. Kim, A. Ortega, J. Ishida, and A. Kubota, "Depth map coding using graph based transform and transform domain sparsification," in *Proc. IEEE Int. Workshop Multimedia Signal*, Oct. 2011, pp. 1–6.

[7] I. Daribo, G. Cheung, and D. Florencio, "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression," in *Proc. 19th IEEE Int. Conf. Image Process.*, Sep. 2012, pp. 1541–1544.

[8] Dynamic-Digital-Depth, Santa Monica, CA, USA. (2012). *Tri-Def3d* [Online]. Available: http://www.tridef.com/

[9] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," *Proc. SPIE*, vol. 5291, pp. 93–104, May 2004.

[10] M. Guttmann, L. Wolf, and D. Cohen-Or, "Semi-automatic stereo extraction from video footage," in *Proc. IEEE ICCV*, Sep./Oct. 2009, pp. 136–142.

[11] I. P. Howard and B. Rogers, *Seeing in Depth*. Oxford, U.K.: Oxford Univ. Press, 2002.

[12] *Study Text of ISO/IEC 14496-10:2008/FPDAM 1 Multiview Video Coding*, document ISO/IEC and JTC1/SC29/WG11, May 2008.

[13] *Advanced Video Coding for Generic Audiovisual Services*, document ITU-T and ISO/IEC-JTC-1, 2010.

[14] S. Kim and Y. Ho, "Mesh-based depth coding for 3D video using hierarchical decomposition of depth maps," in *Proc. IEEE Int. Conf. Image Process.*, Sep./Oct. 2007, pp. V-117–V-120.

[15] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[16] Z. Li, X. Xie, and X. Liu, "An efficient 2D to 3D video conversion method based on skeleton line tracking," in *Proc. 3DTV Conf.*, 2009.

[17] M. Liao, J. Gao, R. Yang, and M. Gong, "Video stereolization: Combining motion analysis with user interaction," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 7, pp. 1079–1088, Jul. 2012.

[18] M. E. Lukacs, "Predictive coding of multi-viewpoint image sets," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Apr. 1986, pp. 521–524.

[19] M. Matthieu and M. N. Do, "Joint encoding of the depth image based representation using shape-adaptive wavelets," in *Proc. IEEE ICIP*, Oct. 2008, pp. 1768–1771.
[20] P. Merkle, C. Bartnik, K. Muller, D. Marpe, and T. Wiegand, "3D video: Depth coding based on inter-component prediction of block partitions," in *Proc. IEEE Picture Coding Symp.*, May 2012, pp. 149–152.
[21] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1461–1473, Nov. 2007.
[22] E. N. Mortensen and W. A. Barrett, "Intelligent scissors for image composition," in *Proc. ACM SIGGRAPH*, 1995.
[23] K.-J. Oh, S. Yea, A. Vetro, and Y.-S. Ho, "Depth reconstruction filter and down/up sampling for depth coding in 3D video," *IEEE Signal Process. Lett.*, vol. 16, no. 9, pp. 747–750, Sep. 2009.
[24] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*. Berlin, Germany: Springer-Verlag, 2005.
[25] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," in *Proc. ACM SIGGRAPH*, 2004, pp. 309–314.
[26] T. Schierl and S. Narasimhan, "Transport and storage systems for 3D video using MPEG-2 systems, RTP, and ISO file format," *Proc. IEEE*, vol. 99, no. 4, pp. 671–683, Apr. 2011.
[27] I. Tabus, I. Schiopu, and J. Astola, "Context coding of depth map images under the piecewise-constant image model representation," *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4195–4210, Nov. 2013.
[28] M. Tanimoto, "Overview of free viewpoint television," *Signal Process. Image Commun.*, vol. 21, no. 6, pp. 454–461, 2006.
[29] V. Nedovic, A. W. M. Smeulders, A. Redert, and J.-M. Geusebroek, "Stages as models of scene geometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1673–1687, Sep. 2010.
[30] M. Wildeboer, T. Yendo, M. Tehrani, T. Fujii, and M. Tanimoto, "Color based depth up-sampling for depth compression," in *Proc. IEEE PCS*, Dec. 2010, pp. 170–173.
[31] C. Wu, G. Er, X. Xie, T. Li, X. Cao, and Q. Dai, "A novel method for semi-automatic 2D to 3D video conversion," in *Proc. 3DTV Conf. True Vis. Capture, Transmiss. Display 3D Video*, May 2008, pp. 65–68.
[32] Z. Zhang, R. Wang, C. Zhou, Y. Wang, and W. Gao, "A compact stereoscopic video representation for 3D video generation and coding," in *Proc. IEEE Data Compress. Conf.*, Apr. 2012, pp. 189–198.
[33] Z. Zhang, Y. Wang, T. Jiang, and W. Gao, "Stereoscopic learning for disparity estimation," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 365–368.
[34] Z. Zhang, Y. Wang, T. Jiang, and W. Gao, "Visual pertinent 2D-to-3D video conversion by multi-cue fusion," in *Proc. IEEE ICIP*, Sep. 2011, pp. 909–912.
[35] Z. Zhang, C. Zhou, Y. Wang, and W. Gao, "Interactive stereoscopic video conversion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1795–1807, Oct. 2013.
[36] Z. Zhang, C. Zhou, B. Xin, Y. Wang, and W. Gao, "An interactive system of stereoscopic video conversion," in *Proc. ACM Multimedia Conf.*, 2012.

Zhebin Zhang is currently a Research Associate with the Department of Electrical Engineering, University of Washington. He received the B.S. degree from the Department of Computer Science and Technology, Beijing University of Posts and Telecommunications, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, in 2005 and 2012, respectively. He held a post-doctoral position with the National Engineering Laboratory for Video Technology, Peking University, from 2012 to 2014. His research interests include computer vision, and image and video process.

Chen Zhou received the B.S. degree from the Computer Science Department, Peking University, in 2011, where he is currently pursuing the Ph.D. degree in computer science. His research interests include computer vision, in particular, 3-D reconstruction.

Ronggang Wang is an Associate Professor with Peking University Shenzhen Graduate School. His research interest is on stereo video and mobile video processing. He has done many technical contributions to the China AVS standard, such as subpel motion compensation, background-predictive picture, and field coding. He has authored about 30 papers in international journals and conferences. He holds more than 20 patents in the field of video coding and processing. He serves as a reviewer of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Signal Processing: Image Communication*, and *GlobalCom*.

Yizhou Wang is a Professor of the Computer Science Department at Peking University (PKU), Beijing, China. He is the Vice Director of the Institute of Digital Media at PKU, and the Director of the New Media Laboratory of the National Engineering Laboratory of Video Technology. He received the bachelor's degree in electrical engineering from Tsinghua University and the Ph.D. degree in computer science from the University of California at Los Angeles in 1996 and 2005, respectively. He was a Research Staff of the Palo Alto Research Center, Xerox, from 2005 to 2008. His research interests include computer vision, statistical modeling and learning, and digital visual arts.

Wen Gao received the Ph.D. degree in electronics engineering from the University of Tokyo, Japan, in 1991. He is a Professor of Computer Science with Peking University, China. Before that, he was a Professor of Computer Science with the Harbin Institute of Technology from 1991 to 1995, and a Professor with the Institute of Computing Technology, Chinese Academy of Sciences. He has authored four books and more than 600 technical articles in refereed journals and conference proceedings in the areas of image processing, video coding and communication, pattern recognition, and multimedia information retrieval. He served on the editorial board for several journals, such as the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE TRANSACTIONS ON MULTIMEDIA, and IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT. He is a fellow of the ACM and a member of CAE.