Quality-Efficient Upsampling Method for Asymmetric Resolution Stereoscopic Video Coding with Interview Motion Compensation and Error Compensation

Kuo-Liang Chung, *Senior Member, IEEE,* Yong-Huai Huang, *Member, IEEE,* and Wen-Chang Liu

***Abstract*—In asymmetric resolution stereoscopic video coding (ARSVC), to reduce the bitrate required for bandwidth-limited channels, each downsampled right-view frame is a quarter the size of the corresponding left-view frame and will be upsampled to the original size by the decoder. In this paper, two upsampling methods for ARSVC are proposed. The first proposed method integrates the traditional Wiener filter-based method and the interview prediction scheme by incorporating the information from the similarity between the left-view and right-view frames. By compensating for the prediction errors, the second proposed method further improves the quality of the upsampled images obtained by the first proposed method, especially in the sequences with heavy irregular textures.**

***Index Terms*—6-tap filter, asymmetric resolution stereoscopic video coding (ARSVC), bitrate reduction, error compensation, interview prediction, three-dimensional television (3-D-TV), upsampling, Wiener filter.**

I. Introduction

NOWADAYS, three-dimensional televisions (3-D-TVs) [1] are becoming increasingly popular in the multimedia and entertainment markets. By synthesizing stereoscopic video sequences, 3-D-TV could provide users with a vivid viewing experience via realistic 3-D scenes. A stereoscopic video sequence consists of two video sequences, one left-view sequence and one synchronized right-view sequence. However, independently encoding the two video sequences results in double storage space and transmission bandwidth requirements. Constrained by limited storage and bandwidth, encoding stereoscopic video sequences with a lower bitrate is therefore crucial [2], [3].

Manuscript received December 31, 2012; revised April 16, 2013, May 27, 2013; accepted July 25, 2013. Date of publication August 6, 2013; date of current version March 4, 2014. This work was supported by the National Science Council of R. O. C. under Contract NSC 101-2221-E-011-139-MY3 and Contract NSC 101-2221-E-228-010. This paper was recommended by Associate Editor R. Hamzaoui.

K.-L. Chung and W.-C. Liu are with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei 10672, Taiwan (klchung01@gmail.com).

Y.-H. Huang is with the Institute of Computer and Communication Engineering and also with the Department of Electronic Engineering, Jinwen University of Science and Technology, Hsin-Tien, New Taipei City 23154, Taiwan (e-mail: yonghuai@ms28.hinet.net).

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

Digital Object Identifier 10.1109/TCSVT.2013.2276877

The suppression theory of our binocular vision system [4] indicates that one view frame of a stereoscopic image pair, say the right-view frame, can be encoded at a lower bitrate than the other view frame, that is, the left-view frame, without visual quality degradation. Subsequently, researchers have proposed asymmetric stereoscopic video coding (ASVC) to realize the above bitrate reduction suggestion for stereoscopic video sequences. According to resolution or quality considerations, ASVC can be classified into two categories; asymmetric resolution stereoscopic video coding (ARSVC) and asymmetric quality stereoscopic video coding (AQSVC). For ARSVC, several methods [5]–[11] have been presented to encode the left-view and right-view video sequences with different resolutions; while for AQSVC, several methods [12]–[17] have been presented to encode stereoscopic video sequences with different quality levels. Both categories provide good opportunities to substantially reduce the bitrate requirement when encoding stereoscopic video sequences [18], [19]. In this paper, we focus on ARSVC and present new quality-efficient upsampling methods for asymmetric stereoscopic video coding by integrating the Wiener filter, interview motion compensation, and error compensation.

In ARSVC, each right-view frame is downsampled to a smaller one while fixing the size of the corresponding left-view frame [5]–[8]. The right-view frame is usually downsampled by a factor of two in both horizontal and vertical directions, resulting in a bitrate reduction effect. After receiving the encoded downsampled right-view frame on the decoder side, the encoded frame is decoded first, and then an upsampling process is necessary to reconstruct the full sized right-view frame. In [5]–[8], each decoded downsampled right-view frame is upsampled by using the 6-tap filter (6TF)-based method proposed in H.264/AVC [20]. Besides the 6TF-based method, the Wiener filter (WF)-based method [21] is an alternative although the method in [21] was originally used to upsample still images. Recently, two improved WF-based upsampling methods, one by Zhang [22] and the other by Zhang *et al.* [23], have been presented. Zhang proposed the soft-decision adaptive interpolation (SAI) method which estimates the missing pixels in a block and jointly trains the relationship not only between the known pixels and the missing pixels but also between the missing pixels themselves

to improve the WF-based method significantly. By connecting the downsampling and upsampling processes, Zhang *et al*. proposed the interpolation-dependent image downsampling with the edge-directed interpolation (IDID-EDI) method to obtain a downsampled frame which contributes to minimize the upsampled error. However, the four existing methods only consider a single image frame without using the relationship between the left-view frame and the corresponding right-view frame. To fully utilize the fact that each right-view frame and the corresponding left-view frame are very much alike, we propose two upsampling methods which utilize the interview prediction scheme, the Wiener filter, and the error compensation scheme to improve the estimation of the missing pixels in the right-view frame.

In this paper, the first proposed upsampling method integrates the interview prediction (IP) scheme and the traditional WF-based method. For simplicity, the proposed method is called the WFIP; it has quality gain when compared with the traditional WF-based method. To realize the proposed WFIP method, two efficient implementations, WFIP_1, adopting the extra-bit recording strategy, and WFIP_2, adopting the fixed-weight strategy, are presented. Next, we incorporate the error compensation scheme into the proposed WFIP method to obtain the proposed WFIP-EC method which can further refine the quality of the upsampled images. Finally, a training-based group merging approach is presented to balance the trade-off between the bitrate reduction and the quality increase for the proposed WFIP-EC method. Experimental results confirm the quality benefit of our proposed upsampling methods for middle and high bitrate cases when compared with the 6TF-based method, the WF-based method, the SAI method, and the IDID-EDI method. The proposed upsampling methods are less competitive, however, for low bitrate cases, but these seldom occur in practice since a low bitrate often leads to visual discomfort in 3-D synthesized scenes.

The rest of this paper is organized as follows. Section II surveys the 6TF-based method and the WF-based method for ARSVC. Section III presents the proposed WFIP method and its two implementations. Section IV presents the proposed improved version of the WFIP and WFIP-EC methods. In Section V, some experimental results are reported to show the quality superiority of the proposed upsampling methods.

II. Two Existing Upsampling Methods for ARSVC: the 6-Tap Filter-Based Method and the WF-Based Method

In this section, two existing upsampling methods, the 6TF-based one and the WF-based one, for ARSVC are briefly presented. Before introducing these two methods, however, we first sketch ARSVC. As shown in Fig. 1, the input stereoscopic video sequence is composed of a left-view video sequence and the synchronized right-view video sequence. In the two sequences, the f-th left-view frame and right-view frame are denoted by L^f and R^f, respectively, each with size $H \times W$. For the f-th right-view frame, the encoder first applies the downsampling process to construct a downsampled right-view frame R^{Df} of size $\frac{H}{2} \times \frac{W}{2}$ and $R^{Df}_{i,j}$ $(= R^f_{2i,2j})$, $0 \le i \le \frac{H}{2} - 1$ and $0 \le j \le \frac{W}{2} - 1$, denotes the pixel at location (i, j) in the downsampled right-view frame. Instead of encoding R^f, the encoder encodes R^{Df} to obtain the bitrate reduction effect.

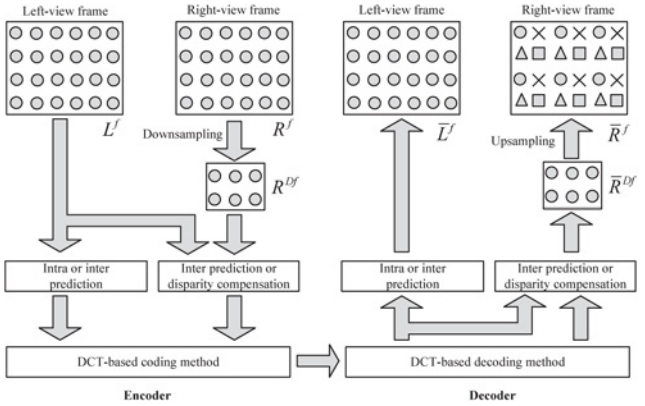


Fig. 1. Flowchart for ARSVC.

In ARSVC, the encoding strategies for L^f and R^{Df} are somewhat different. For L^f, each current block is predicted either by inter prediction or by intra prediction in the joint multiview video coding (JMVC) standard [25]; in inter prediction, the motion compensation is applied to the current block by referring to the previous left-view frame. In order to remove the redundancy between the two views in a stereoscopic frame pair, the disparity compensation [24] or inter prediction is used to predict R^{Df} by referring to either the downsampled version of L^f or the previous downsampled right-view frame, respectively. After encoding the prediction errors of L^f and R^{Df} by the discrete cosine transformation-based (DCT-based) technique, the encoded stereoscopic frame pair is transmitted to the decoder side for displaying the 3-D scene.

A. 6-Tap Filter-Based Upsampling Method

Based on the 6TF-based upsampling method used in H.264/AVC, the decoder first decodes the encoded left-vew frame and downsampled right-view frame to obtain $\bar{L}^f$ and $\bar{R}^{Df}$, respectively, and then reconstructs the full-sized right-view frame $\bar{R}^f$ from $\bar{R}^{Df}$. As shown in Fig. 1, the circle-marked pixels in $\bar{R}^f$ are duplicated from $\bar{R}^{Df}$ and we have $\bar{R}^f_{2i,2j} = \bar{R}^{Df}_{i,j}$, $0 \le i \le \frac{H}{2} - 1$ and $0 \le j \le \frac{W}{2} - 1$. Except for the known circle-marked pixels, the cross-marked pixels, triangle-marked pixels, and square-marked pixels, which are located at positions $(2i, 2j + 1)$, $(2i + 1, 2j)$, and $(2i + 1, 2j + 1)$ in $\bar{R}^f$, respectively, are the missing pixels to be estimated by the 6TF-based method. It first interpolates all $\bar{R}^f_{2i,2j+1}$s and $\bar{R}^f_{2i+1,2j}$s, that is, the cross-marked pixels and the triangle-marked pixels, respectively, by computing

$$\bar{R}^f_{2i,2j+1} = \frac{1}{32}\bar{R}^f_{2i,2j-4} - \frac{5}{32}\bar{R}^f_{2i,2j-2} + \frac{20}{32}\bar{R}^f_{2i,2j} + \frac{20}{32}\bar{R}^f_{2i,2j+2} - \frac{5}{32}\bar{R}^f_{2i,2j+4} + \frac{1}{32}\bar{R}^f_{2i,2j+6} \quad (1)$$

and

$$\bar{R}^f_{2i+1,2j} = \frac{1}{32}\bar{R}^f_{2i-4,2j} - \frac{5}{32}\bar{R}^f_{2i-2,2j} + \frac{20}{32}\bar{R}^f_{2i,2j} + \frac{20}{32}\bar{R}^f_{2i+2,2j} - \frac{5}{32}\bar{R}^f_{2i+4,2j} + \frac{1}{32}\bar{R}^f_{2i+6,2j}. \quad (2)$$

Finally, making use of the known circle-marked, cross-marked, and triangle-marked pixels, the square-marked pixels are interpolated by

$$\bar{R}^f_{2i+1,2j+1} = \frac{F^{Hf}_{2i+1,2j+1} + F^{Vf}_{2i+1,2j+1}}{2} \tag{3}$$

where

$$\begin{aligned} F^{Hf}_{2i+1,2j+1} = {} & \frac{1}{32}\bar{R}^f_{2i+1,2j-4} - \frac{5}{32}\bar{R}^f_{2i+1,2j-2} + \\ & \frac{20}{32}\bar{R}^f_{2i+1,2j} + \frac{20}{32}\bar{R}^f_{2i+1,2j+2} - \\ & \frac{5}{32}\bar{R}^f_{2i+1,2j+4} + \frac{1}{32}\bar{R}^f_{2i+1,2j+6} \end{aligned} \tag{4}$$

and

$$\begin{aligned} F^{Vf}_{2i+1,2j+1} = {} & \frac{1}{32}\bar{R}^f_{2i-4,2j+1} - \frac{5}{32}\bar{R}^f_{2i-2,2j+1} + \\ & \frac{20}{32}\bar{R}^f_{2i,2j+1} + \frac{20}{32}\bar{R}^f_{2i+2,2j+1} - \\ & \frac{5}{32}\bar{R}^f_{2i+4,2j+1} + \frac{1}{32}\bar{R}^f_{2i+6,2j+1}. \end{aligned} \tag{5}$$

B. Wiener Filter-Based Upsampling Method

In this subsection, the WF-based upsampling method [21] is introduced. For each missing pixel, if the variance of its four nearest neighboring pixels is larger than the threshold, say eight empirically, the missing pixel is predicted to be an edge pixel; otherwise, it is predicted to be a non-edge pixel. The bilinear interpolation method is used to estimate each missing non-edge pixel. The WF-based method is applied to estimate each missing edge pixel. Among the three kinds of unknown pixels in Fig. 1, the WF-based upsampling method first interpolates the square-marked pixels $\bar{R}^f_{2i+1,2j+1}$s by referring to the four existing decoded circle-marked pixels, $\bar{R}^f_{2i,2j}$, $\bar{R}^f_{2i,2j+2}$, $\bar{R}^f_{2i+2,2j}$, and $\bar{R}^f_{2i+2,2j+2}$, using the least square technique. For easy exposition, the index pair $(2i+1, 2j+1)$ is replaced by (m, n). Thus $\bar{R}^f_{m,n}$ is estimated by

$$\begin{aligned} \bar{R}^f_{m,n} = {} & \alpha_0\bar{R}^f_{m-1,n-1} + \alpha_1\bar{R}^f_{m-1,n+1} + \alpha_2\bar{R}^f_{m+1,n-1} \\ & + \alpha_3\bar{R}^f_{m+1,n+1} \end{aligned} \tag{6}$$

where the four unknown coefficients, α_0, α_1, α_2, and α_3, are determined by the following window-based least square technique. As shown in Fig. 2, a 7×7 $(=(3\times2+1)\times(3\times2+1))$ window is adopted and the number of reference circle-marked pixels is sixteen; these reference pixels $\bar{R}^f_{m+k,n+l}$s, where k and l are odd and $-3 \le k, l \le 3$, are rearranged as a 16×1 vector $\vec{y}$. The vector $\vec{y}$ is put into (7), in which each entry $\bar{R}^f_{m+k,n+l} \in \vec{y}$ is equal to the inner product of the four pixels, $\bar{R}^f_{m+k-2,n+l-2}$, $\bar{R}^f_{m+k-2,n+l+2}$, $\bar{R}^f_{m+k+2,n+l-2}$, and $\bar{R}^f_{m+k+2,n+l+2}$, and the four unknown coefficients.

$$\arg\min_{\vec{\alpha}} \|\vec{y} - C\vec{\alpha}\|^2 \tag{7}$$

where

$$\vec{y} = \left[\, \bar{R}^f_{m-3,n-3} \;\; \bar{R}^f_{m-3,n-1} \;\cdots\; \bar{R}^f_{m+3,n+3} \,\right]^T \tag{8}$$

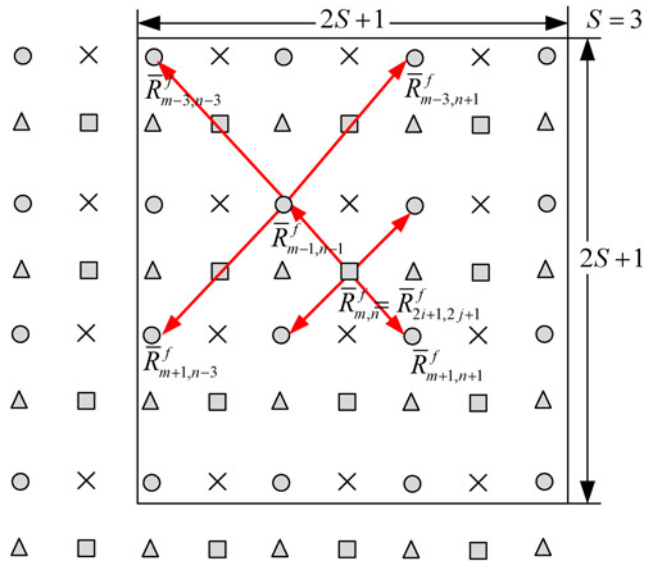


Fig. 2. Reference circle-marked pixels for estimating square-marked pixels $\bar{R}^f_{2i+1,2j+1}$s.

$$C = \begin{bmatrix} \bar{R}^f_{m-5,n-5} & \bar{R}^f_{m-5,n-1} & \bar{R}^f_{m-1,n-5} & \bar{R}^f_{m-1,n-1} \\ \bar{R}^f_{m-5,n-3} & \bar{R}^f_{m-5,n+1} & \bar{R}^f_{m-5,n+1} & \bar{R}^f_{m-1,n+1} \\ \vdots & \vdots & \vdots & \vdots \\ \bar{R}^f_{m+1,n+1} & \bar{R}^f_{m+1,n+5} & \bar{R}^f_{n+5,n+1} & \bar{R}^f_{m+5,n+5} \end{bmatrix} \tag{9}$$

and

$$\vec{\alpha} = \left[\, \alpha_0 \;\; \alpha_1 \;\; \alpha_2 \;\; \alpha_3 \,\right]^T. \tag{10}$$

Following the least square technique, the four unknown coefficients are determined by

$$\vec{\alpha} = (C^T C)^{-1}(C^T \vec{y}). \tag{11}$$

By (6), the square-marked pixels are thus estimated. We now proceed to estimate the cross-marked pixels. For convenience, the index pair $(2i, 2j+1)$ is replaced by (m, n). As shown in Fig. 3, the missing cross-marked pixel $\bar{R}^f_{m,n}$ can be estimated with the help of the two circle-marked pixels, $\bar{R}^f_{m,n+1}$ and $\bar{R}^f_{m,n-1}$, and the two square-marked pixels, $\bar{R}^f_{m-1,n}$ and $\bar{R}^f_{m+1,n}$. It yields

$$\bar{R}^f_{m,n} = \beta_0\bar{R}^f_{m-1,n} + \beta_1\bar{R}^f_{m,n-1} + \beta_2\bar{R}^f_{m,n+1} + \beta_3\bar{R}^f_{m+1,n}. \tag{12}$$

Similar to solving (7), we can solve the four unknown coefficients in (12) by solving the related overdetermined system which is constructed from the sixteen reference pixels marked by circles and squares in the $(S+1)\times(S+1)$ diamond window, as shown in Fig. 3.

III. First Proposed Wiener Filter- and Interview Prediction-Based Upsampling Method: The WFIP Method

This section presents the first upsampling method which integrates the Wiener filter and interview prediction scheme. It is called WFIP method, and involves a hybrid approach combining the WF technique and the IP scheme. Especially,

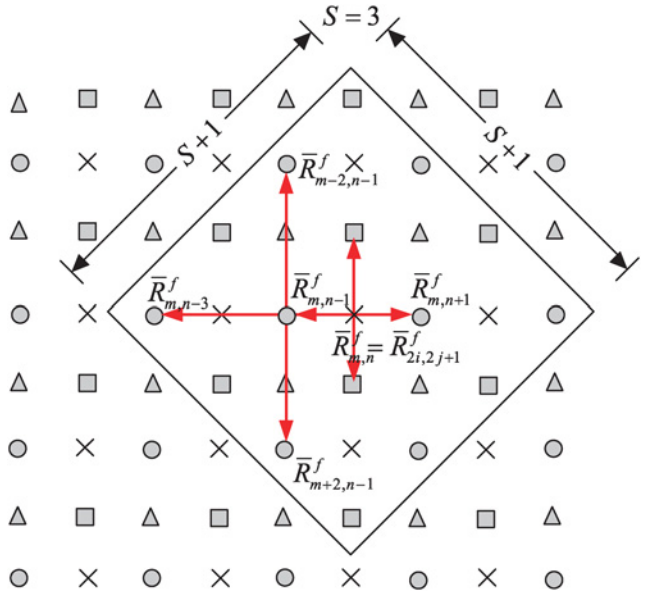


Fig. 3. Reference circle-marked and square-marked pixels for estimating cross-marked pixels $\bar{R}^f_{2i,2j+1}$s.

the left-view frame and the downsampled right-view frame are used together as the training data during the upsampling process. In order to take the interview correlation into account, $\bar{R}^f$ is first partitioned into a set of non-overlapping 16×16 blocks $B^f_{a,b}$s, $0 \le a \le \frac{H}{16} - 1$ and $0 \le b \le \frac{W}{16} - 1$, in which for each block $B^f_{a,b}$, the pixel in the upper-left corner is denoted by $\bar{R}^f_{16a,16b}$. For each $B^f_{a,b}$ in $\bar{R}^f$, the best matched block in a $(2D+1) \times (2D+1)$ search window of $\bar{L}^f$ can be characterized by the motion vector $(x^f_{a,b}, y^f_{a,b})$ in (13), which minimizes the sum of absolute differences between $B^f_{a,b}$ and the reference block in the search window

$$(x^f_{a,b}, y^f_{a,b}) = \arg \min_{-D \le x,y \le D} f(x, y) \tag{13}$$

where

$$f(x, y) = \sum_{k=0}^{7} \sum_{l=0}^{7} |\bar{R}^f_{16a+2k,16b+2l} - \bar{L}^f_{16a+x+2k,16b+y+2l}|. \tag{14}$$

Here, D is set to 20 empirically. Note that in $B^f_{a,b}$, only the decoded pixels $\bar{R}^f_{2i,2j}$s, $0 \le i \le \frac{H}{2}$ and $0 \le j \le \frac{W}{2}$, are considered in the above block matching process.

For each partitioned 16×16 block in $\bar{R}^f$, after performing the above block matching process, the best matched block in $\bar{L}^f$ has been found. Because $\bar{L}^f$ is of full size, a smaller window with size $(S+1) \times (S+1)$ is adopted to cover the training data instead of adopting the window with size $(2S+1) \times (2S+1)$ mentioned in Fig. 2. Fig. 4 shows a 4×4 window and the covered training data contains sixteen pixels $\bar{L}^f_{m+k,n+l}$s, $-1 \le k, l \le 2$. The resultant overdetermined system is given by

$$\arg \min_{\vec{\alpha_L}} \|\vec{y_L} - C_L \vec{\alpha_L}\|^2 \tag{15}$$

where

$$\vec{y_L} = \left[\bar{L}^f_{m-1,n-1} \; \bar{L}^f_{m-1,n} \cdots \bar{L}^f_{m+2,n+2} \right]^T \tag{16}$$

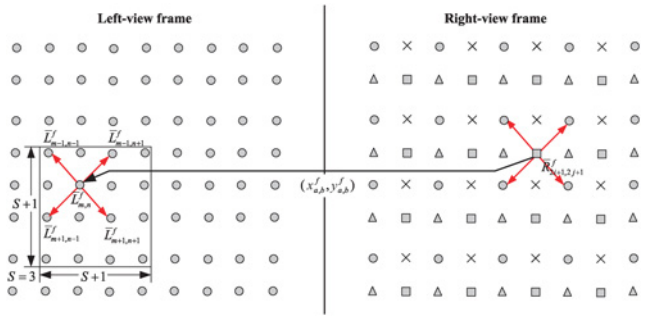


Fig. 4. Interview correlation for estimating the missing pixel $\bar{R}^f_{2i+1,2j+1}$.

$$C_L = \begin{bmatrix} \bar{L}^f_{m-2,n-2} & \bar{L}^f_{m-2,n} & \bar{L}^f_{m,n-2} & \bar{L}^f_{m,n} \\ \bar{L}^f_{m-2,n-1} & \bar{L}^f_{m-2,n+1} & \bar{L}^f_{m,n-1} & \bar{L}^f_{m,n+1} \\ \vdots & \vdots & \vdots & \vdots \\ \bar{L}^f_{m+1,n+1} & \bar{L}^f_{m+1,n+3} & \bar{L}^f_{m+3,n+1} & \bar{L}^f_{m+3,n+3} \end{bmatrix} \tag{17}$$

$$\vec{\alpha_L} = \left[\alpha_0 \; \alpha_1 \; \alpha_2 \; \alpha_3 \right]^T. \tag{18}$$

$\vec{\alpha_L}$ can be obtained by

$$\vec{\alpha_L} = (C_L^T C_L)^{-1} (C_L^T \vec{y_L}). \tag{19}$$

The determined four coefficients are thus used to estimate the square-marked pixels. Further, we estimate the cross-marked pixels in the same way, and finally estimate the triangle-marked pixels.

When estimating each missing pixel, its estimated value may be obtained by the WF-based method or by the proposed IP-based method. In what follows, we present two implementations to resolve this controversy. In the first implementation, called the WFIP_1 method, for each pixel, the encoder compares its two estimation errors, one generated by the WF-based method and the other generated by the proposed IP-based method, and then uses an extra bit to record which method is preferred for estimating the missing pixel. On the other hand, according to the recorded extra bit, for each missing pixel, the decoder easily knows which one, the WF-based method or the proposed IP-based method, should be applied to estimate the missing pixel. To achieve a compromise between the bitrate and the quality, a block-based strategy is suggested and each missing pixel in the 16×16 block shares one common favorite extra bit. To record the extra-bit bitstream, we first apply the WF-based method and the proposed IP-based method to upsample R^{Df}, respectively, to obtain two full sized right-view frames. We then calculate the two estimation error maps between the original right-view frame and the upsampled right-view frames generated by the WF-based method and the proposed IP-based method, respectively. Based on the two estimation error maps, for each block, an extra bit 1 is recorded if the WF-based method is better than the proposed IP-based method; otherwise bit 0 is recorded. Fig. 5 depicts the two implementations, the WFIP_1 and WFIP_2 methods, for the proposed WFIP method. As shown in Fig. 5(a), the encoder encodes L^f and R^{Df} as a bitstream BS^f_E and then sends it and the extra-bit bitstream BS^f_X to the decoder. As shown

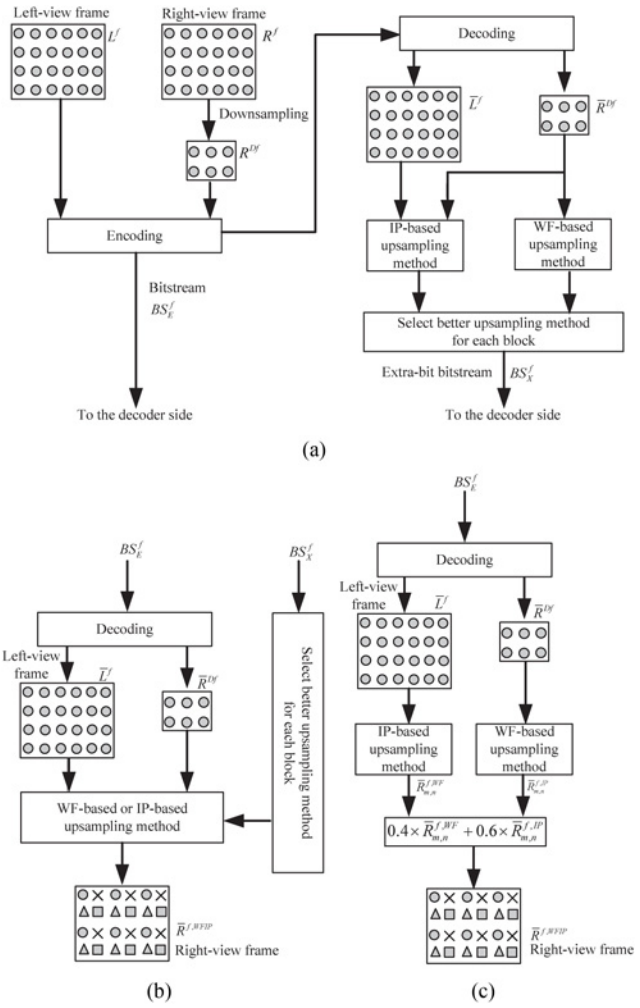


Fig. 5. Two implementations, the WFIP_1 and WFIP_2 methods, for the proposed WFIP method. (a) Encoder side of the WFIP_1 method. (b) Decoder side of the WFIP_1 method. (c) Decoder side of the WFIP_2 method.

in Fig. 5(b), after receiving the two bitstreams, the decoder reconstructs the left-view frame $\bar{L}^f$ and the downsampled right-view frame $\bar{R}^{Df}$ according to the bitstream BS_E^f and then upsamples $\bar{R}^{Df}$ to the full sized $\bar{R}^f$ according to BS_X^f. Our experiments show that the average PSNR improvement of the proposed WFIP_1 method is 0.70 dB when compared with the WF-based method; the average bitrate overhead is 6.16% when the values of quantization parameters (QPs) are set to 16, 20, 24, 28, 32, 36, and 40.

After examining BS_X^f, the ratio of the number of 0s over the number of 1s is 3/2. Therefore, each missing pixel in the right-view frame can be estimated by

$$\bar{R}_{m,n}^{f,WFIP} = 0.6 \times \bar{R}_{m,n}^{f,IP} + 0.4 \times \bar{R}_{m,n}^{f,WF} \tag{20}$$

where $\bar{R}_{m,n}^{f,WF}$ and $\bar{R}_{m,n}^{f,IP}$ denote, respectively, the upsampled pixels obtained by the WF-based method and the proposed IP-based method. Different from the first implementation which performs the upsampling process on both the encoder and decoder sides, the second implementation only performs the upsampling process on the decoder side. Under seven testing stereoscopic video sequences for QP=16, 20, 24, 28, 32, 36, and 40, the average PSNR improvement of the proposed WFIP_2 method over the WF-based method is 0.50 dB; the proposed WFIP_2 method has no bitrate overhead.

IV. Second Proposed Wfip- and Error-Compensation-Based Upsampling Method: The Wfip-EC Method

In this section, the second proposed upsampling method, the WFIP- and error compensation-based (WFIP-EC) method, is presented. In the proposed WFIP-EC method, we improve the proposed WFIP method by adopting the error compensation scheme to further refine the quality.

After performing the proposed WFIP method, let the estimated value of each square-marked pixel be denoted by $\bar{R}_{m,n}^{f,WFIP}$, $(m,n) \in (2i+1, 2j+1)$, and the corresponding estimation error be $\bar{E}_{m,n}^{f,WFIP}$. The refined square-marked pixel can be obtained by

$$\bar{R}_{m,n}^{f,WFIP-EC} = \bar{R}_{m,n}^{f,WFIP} + \bar{E}_{m,n}^{f,WFIP}. \tag{21}$$

Following the same correlation assumption discussed in the WF-based method, the error compensation term $\bar{E}_{m,n}^{f,WFIP}$ can be estimated by the four estimation errors, $\bar{E}_{m-1,n-1}^{f,WFIP}$, $\bar{E}_{m-1,n+1}^{f,WFIP}$, $\bar{E}_{m+1,n-1}^{f,WFIP}$, and $\bar{E}_{m+1,n+1}^{f,WFIP}$, of the four circle-marked pixels in the right-view frame, and the four estimation errors can be calculated by

$$\bar{E}_{k,l}^{f,WFIP} = \bar{R}_{k,l}^{f} - \bar{R}_{k,l}^{f,WFIP} \tag{22}$$

where $(k,l) \in \{(m-1,n-1), (m-1,n+1), (m+1,n-1), (m+1,n+1)\}$; as shown in Fig. 6(a), $\bar{R}_{k,l}^{f,WFIP}$ is calculated using the proposed WFIP method by referring to the four neighboring circle-marked pixels $\bar{R}_{k-2,l-2}^{f}$, $\bar{R}_{k-2,l+2}^{f}$, $\bar{R}_{k+2,l-2}^{f}$, and $\bar{R}_{k+2,l+2}^{f}$.

As shown in Fig. 6(b), after calculating the four error terms, $\bar{E}_{k,l}^{f,WFIP}$ for $(k,l) \in \{(m-1,n-1), (m-1,n+1), (m+1,n-1), (m+1,n+1)\}$, the unknown error term $\bar{E}_{m,n}^{f,WFIP}$ of the squared-marked pixel can be estimated by

$$\begin{aligned}\bar{E}_{m,n}^{f,WFIP} = {} & \gamma_0 \bar{E}_{m-1,n-1}^{f,WFIP} + \gamma_1 \bar{E}_{m-1,n+1}^{f,WFIP} + \\ & \gamma_2 \bar{E}_{m+1,n-1}^{f,WFIP} + \gamma_3 \bar{E}_{m+1,n+1}^{f,WFIP}.\end{aligned} \tag{23}$$

The above error compensation scheme can be applied to estimate the related error compensation terms of the cross-marked pixels and triangular-marked pixels. Because the decoder does not have the original values of the square-marked pixels, the corresponding estimation errors cannot actually be computed. Therefore, we propose a training-based approach on the encoder side to determine the four coefficients γ_0, γ_1, γ_2, and γ_3. Instead of sending four new coefficients to the decoder for estimating the error compensation term of each square-marked pixel, the encoder only needs to send twelve coefficients in total to the decoder for refining the quality of the upsampled right-view frame. In what follows, the details of the training process are described.

Let the four neighboring error terms of the current square-marked pixel be called the error pattern. After collecting all these error patterns as a training set, the encoder first classifies them into seven groups, and then they are further merged into three compact groups by the affine mapping technique.

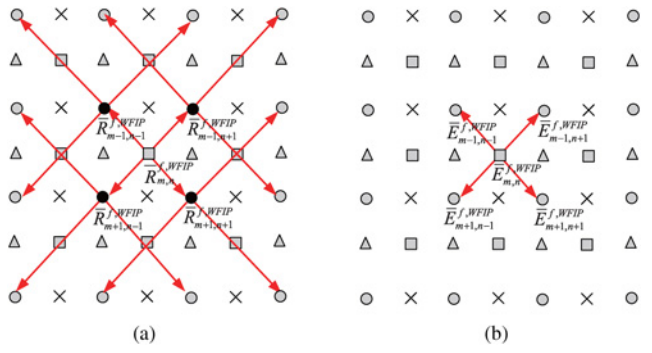


Fig. 6. Depiction of the proposed error compensation scheme. (a) Computing error terms of the four decoded circle-marked pixels. (b) Estimating the error of the square-marked pixel using the four neighboring error terms.

It is known that the error pattern $P^f_{m,n}$ consists of $\bar{E}^{f,WFIP}_{m-1,n-1}$, $\bar{E}^{f,WFIP}_{m-1,n+1}$, $\bar{E}^{f,WFIP}_{m+1,n-1}$, and $\bar{E}^{f,WFIP}_{m+1,n+1}$. According to the gradient direction and magnitude of each training error pattern, the following classification rule is used to determine the group index that the error pattern $P^f_{m,n}$ belongs to

$$P^f_{m,n} \in \begin{cases} G_1, \text{ if } (|\bar{E}^{f,WFIP}_{m-1,n-1}| + |\bar{E}^{f,WFIP}_{m+1,n+1}|) - \\ \qquad (|\bar{E}^{f,WFIP}_{m-1,n+1}| + |\bar{E}^{f,WFIP}_{m+1,n-1}|) > T_G \\ G_2, \text{ if } (|\bar{E}^{f,WFIP}_{m-1,n+1}| + |\bar{E}^{f,WFIP}_{m+1,n-1}|) - \\ \qquad (|\bar{E}^{f,WFIP}_{m-1,n-1}| + |\bar{E}^{f,WFIP}_{m+1,n+1}|) > T_G \\ G_3, \text{ if } (|\bar{E}^{f,WFIP}_{m-1,n-1}| + |\bar{E}^{f,WFIP}_{m-1,n+1}|) - \\ \qquad (|\bar{E}^{f,WFIP}_{m+1,n-1}| + |\bar{E}^{f,WFIP}_{m+1,n+1}|) > T_G \\ G_4, \text{ if } (|\bar{E}^{f,WFIP}_{m+1,n-1}| + |\bar{E}^{f,WFIP}_{m+1,n+1}|) - \\ \qquad (|\bar{E}^{f,WFIP}_{m-1,n-1}| + |\bar{E}^{f,WFIP}_{m-1,n+1}|) > T_G \\ G_5, \text{ if } (|\bar{E}^{f,WFIP}_{m-1,n-1}| + |\bar{E}^{f,WFIP}_{m+1,n-1}|) - \\ \qquad (|\bar{E}^{f,WFIP}_{m-1,n+1}| + |\bar{E}^{f,WFIP}_{m+1,n+1}|) > T_G \\ G_6, \text{ if } (|\bar{E}^{f,WFIP}_{m-1,n+1}| + |\bar{E}^{f,WFIP}_{m+1,n+1}|) - \\ \qquad (|\bar{E}^{f,WFIP}_{m-1,n-1}| + |\bar{E}^{f,WFIP}_{m+1,n-1}|) > T_G \\ G_7, \quad \text{otherwise} \end{cases} \tag{24}$$

where the threshold T_G is set to 17 empirically. In (24), the error pattern $P^f_{m,n} \in \{G_1, G_2\}$ indicates that $P^f_{m,n}$ belongs to a diagonal group because $P^f_{m,n}$ has a diagonal texture; $P^f_{m,n} \in \{G_3, G_4\}$ or $P^f_{m,n} \in \{G_5, G_6\}$ implies that $P^f_{m,n}$ belongs to a horizontal group or vertical group, respectively, because $P^f_{m,n}$ has a horizontal texture or vertical texture; otherwise, $P^f_{m,n}$ is classified to G_7.

In what follows, an affine transformation technique is adopted to merge the above seven groups into three compact groups. Since the error pattern in G_1 can be transferred to that in G_2 by horizontal mirror mapping and vice versa, we thus merge G_1 and G_2 into a new group, say G_a. G_3 and G_4 can be merged into a new group, say G_b, by vertical mirror mapping. Furthermore, since we can map the error pattern in G_3 to be that in G_5 or G_6 by rotating the error pattern in G_3 by 90 or -90 degrees, respectively, we thus merge G_3, G_4, G_5, and G_6 into a new group, say G_b. For convenience, G_7 is renamed to G_c. Based on the three compact groups, we collect all error patterns belonging to the same group, then the least square method is applied to these collected error patterns in the same group G_g, $g \in \{a, b, c\}$, and the four unknown coefficients, γ^g_0, γ^g_1, γ^g_2, and γ^g_3, can be determined. Based on

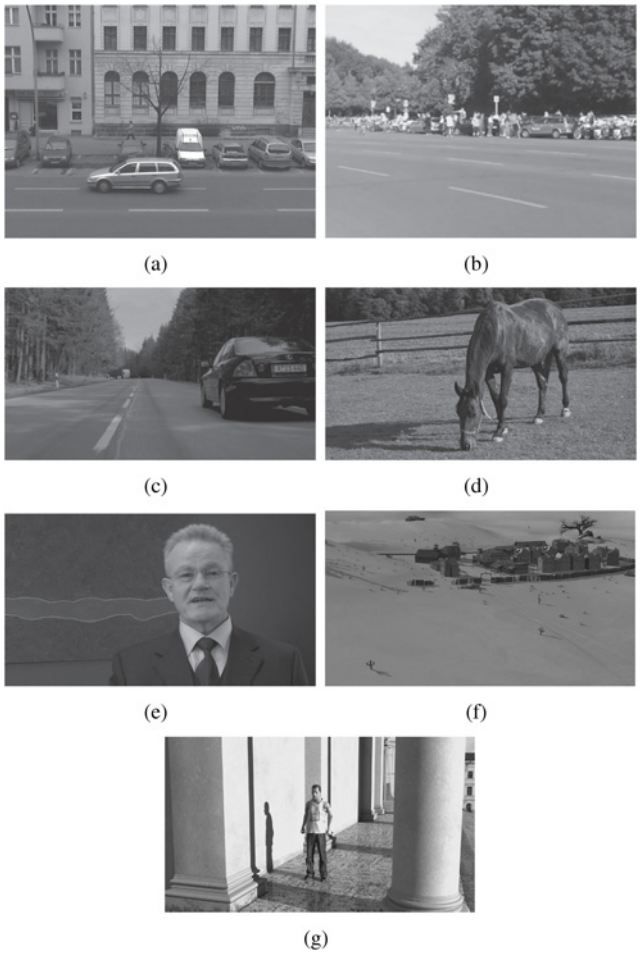


Fig. 7. First right-view frames. (a) *Alt Moabit* sequence. (b) *Rollerblade* sequence. (c) *Car* sequence. (d) *Horse* sequence. (e) *Bullinger* sequence. (f) *GT Fly* sequence. (g) *Undo Dancer* sequence.

the above group merging technique, instead of sending four new coefficients to the decoder for refining the estimation of each missing pixel in the right-view frame, the encoder only needs to send twelve coefficients in total to the decoder to refine the estimation of all missing pixels. After receiving the twelve coefficients, the decoder builds up a table to record them. To refine the estimation of each missing square-marked pixel, the decoder will select the proper four coefficients from the table based on the group type that the error pattern of the current missing pixel belongs to. Next, (23) is used to calculate the error compensation term. Further, (21) is applied to refine the estimation quality of each missing square-marked pixel. By the same arguments, a similar error compensation process can be applied to the quality refinement of upsampled cross-marked pixels and triangular-marked pixels. The above proposed training-based group merging approach not only leads to a significant bitrate reduction effect, but the experimental results also demonstrate a sufficient quality refinement.

V. Experimental Results

In this section, seven test stereoscopic video sequences are downloaded from the website [26] to compare the performance among the 6TF-based method [20], the WF-based

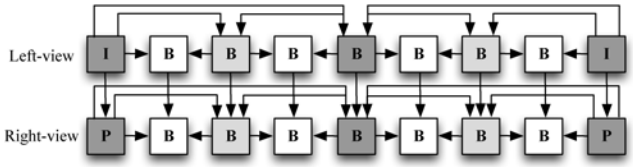


Fig. 8. GOP structure used in our experiments.

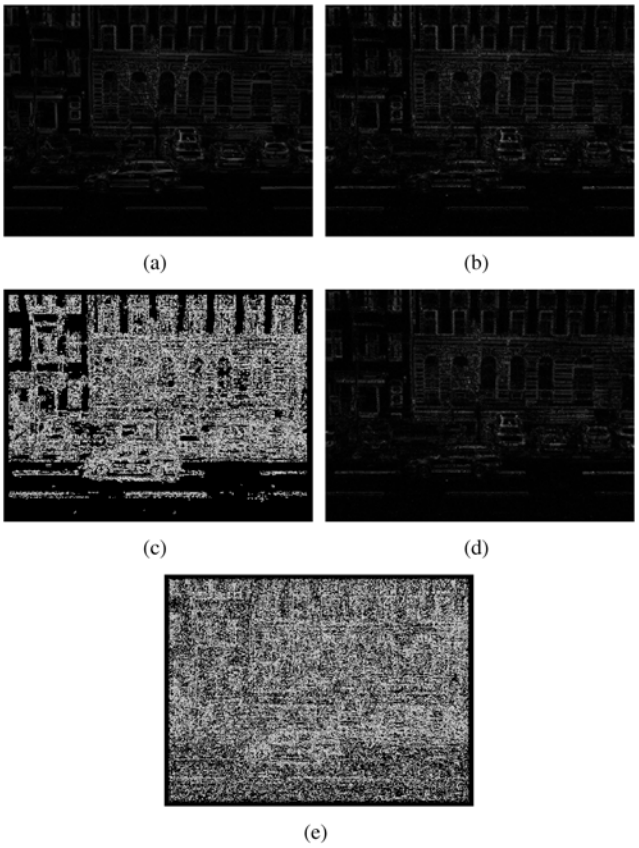


Fig. 9. Quality benefit of the proposed WFIP_1 and WFIP-EC_1 methods over the WF-based method for the *Alt Moabit* video sequence. (a) and (b) are the residuals between Fig. 7(a) and the upsampled frames generated by the WF-based method and the proposed WFIP_1 method, respectively. (c) Better upsampled pixels in white using the proposed WFIP_1 method. (d) Residual between Fig. 7(a) and the upsampled frame generated by the proposed WFIP-EC_1 method. (e) The better upsampled pixels in white using the proposed WFIP-EC_1 method.

method [21], the SAI method [22], the IDID-EDI method [23], the proposed WFIP methods, and the proposed WFIP-EC methods for ARSVC. The performance comparison is evaluated by the two terms, the bitrate and the quality in terms of PSNR. In our experiments, the seven test video sequences are the *Alt Moabit* sequence, each frame with size 512 × 384, the *Rollerblade* sequence, each frame with size 320 × 240, the *Car*, *Horse*, and *Bullinger* sequences, each frame with size 432 × 240, and the *GT Fly* and *Undo Dancer* sequences, each frame with size 1920 × 1088. Fig. 7 demonstrates the first right-view frames of the seven test sequences. All experiments were implemented on an IBM compatible computer with an Intel Core i7 3770 CPU 3.40 GHz and 16GB RAM. The operating system was Microsoft

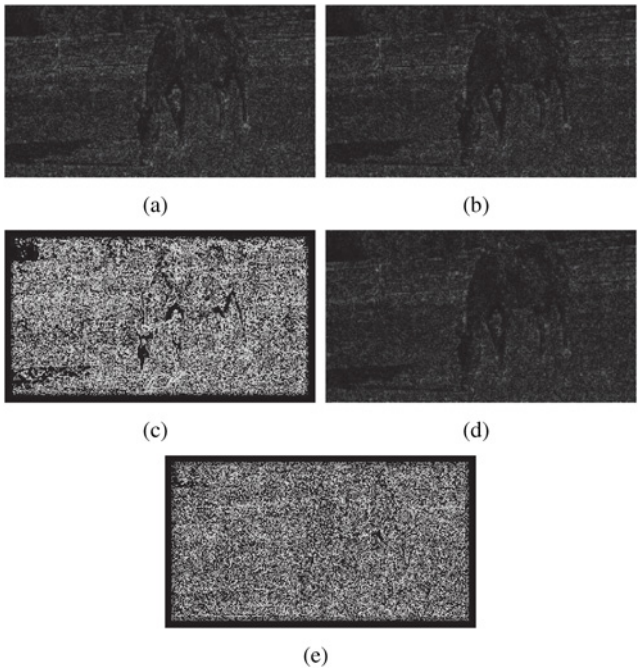


Fig. 10. Quality benefit of the proposed WFIP_1 and WFIP-EC_1 methods over the WF-based method for the *Horse* video sequence. (a) and (b) are the residuals between Fig. 7(d) and the upsampled frames generated by the WF-based method and the proposed WFIP_1 method, respectively. (c) Better upsampled pixels in white using the proposed WFIP_1 method. (d) Residual between Fig. 7(d) and the upsampled frame generated by the proposed WFIP-EC_1 method. (e) The better upsampled pixels in white using the proposed WFIP-EC_1 method.

Windows 7; the program development environment was Visual C++ 2005; the implementation platform was JMVC 8.5. The GOP size was set to 8 and its structure is shown in Fig. 8. Seven different QPs, 16, 20, 24, 28, 32, 36, and 40 are selected for encoding the seven test sequences. The parameter S related to window size for constructing the overdetermined system is set to 7 and the radius D related to the search window in the block matching process is set to 20. In what follows, three kinds of experiments are carried out to demonstrate the quality superiority of our proposed upsampling methods.

The first experiment demonstrates the quality improvement of the proposed WFIP and WFIP-EC methods over the WF-based method. As shown in Fig. 9, we only show the quality benefit of the proposed WFIP_1 and WFIP-EC_1 methods because that of the proposed WFIP_2 and WFIP-EC_2 methods is the same. Given a right-view frame as shown in Fig. 7(a), for QP =16, Fig. 9(a) illustrates the residual between Fig. 7(a) and the upsampled frame by using the WF-based method. Fig. 9(b) illustrates the residual between Fig. 7(a) and the upsampled one by using the proposed WFIP_1 method. As shown in Fig. 9(c), the better upsampled pixels produced by the proposed WFIP_1 method are depicted by the white pixels, revealing the sufficient quality improvement of the proposed WFIP_1 method over the WF-based method. Fig. 9(d) illustrates the corresponding residual by using the proposed WFIP-EC_1 method. From the difference between Fig. 9(b) and Fig. 9(d), that is, Fig. 9(e), a considerable number of better upsampled pixels produced by the proposed WFIP-

TABLE I

QUALITY AND BITRATE COMPARISON FOR THE ALT MOABIT VIDEO SEQUENCE FOR PSNR (DB) AND KILOBITS PER SECOND, RESPECTIVELY

Method \ QP		16	20	24	28	32	36	40	Average
6TF-based	PSNR	30.24	30.18	30.01	29.67	29.00	27.94	26.60	29.09
	Bitrate	2205	1463	978	643	412	264	166	876
WF-based	PSNR	30.40	30.33	30.16	29.80	29.11	28.02	26.62	29.20
	Bitrate	2205	1463	978	643	412	264	166	876
SAI	PSNR	30.68	30.62	30.38	29.90	29.16	28.00	26.62	29.34
	Bitrate	2205	1463	978	643	412	264	166	876
IDID-EDI	PSNR	30.41	30.39	30.30	30.07	29.47	28.32	26.74	29.39
	Bitrate	2388	1603	1080	716	453	281	171	956
WFIP_1	PSNR	31.83	31.57	31.15	30.50	29.54	28.27	26.75	29.95
	Bitrate	2228	1486	1001	666	435	287	189	899
WFIP_2	PSNR	31.48	31.28	30.90	30.25	29.24	28.02	26.51	29.67
	Bitrate	2205	1463	978	643	412	264	166	876
WFIP-EC_1	PSNR	31.87	31.62	31.19	30.55	29.58	28.29	26.76	29.98
	Bitrate	2240	1497	1012	678	447	299	201	911
WFIP-EC_2	PSNR	31.56	31.35	30.96	30.30	29.28	28.03	26.51	29.71
	Bitrate	2217	1474	989	655	424	276	178	887

TABLE II

QUALITY AND BITRATE COMPARISON FOR THE ROLLERBLADE VIDEO SEQUENCE FOR PSNR (DB) AND KILOBITS PER SECOND, RESPECTIVELY

Method \ QP		16	20	24	28	32	36	40	Average
6TF-based	PSNR	29.66	29.59	29.41	29.05	28.27	26.67	24.61	28.18
	Bitrate	1111	662	395	235	140	88	58	384
WF-based	PSNR	29.05	28.95	28.77	28.43	27.73	26.32	24.44	27.67
	Bitrate	1111	662	395	235	140	88	58	384
SAI	PSNR	29.69	29.52	29.24	28.89	28.11	26.56	24.59	28.08
	Bitrate	1111	662	395	235	140	88	58	384
IDID-EDI	PSNR	29.13	29.04	28.87	28.58	27.97	26.66	24.71	27.85
	Bitrate	1294	802	497	307	181	105	63	464
WFIP_1	PSNR	30.35	30.19	29.90	29.39	28.44	26.74	24.66	28.53
	Bitrate	1134	685	418	258	163	111	81	407
WFIP_2	PSNR	30.01	29.86	29.60	29.14	28.26	26.63	24.59	28.30
	Bitrate	1111	662	395	235	140	88	58	384
WFIP-EC_1	PSNR	30.52	30.36	30.06	29.54	28.57	26.83	24.70	28.65
	Bitrate	1146	696	429	269	175	123	93	419
WFIP-EC_2	PSNR	30.29	30.14	29.88	29.39	28.46	26.75	24.64	28.51
	Bitrate	1123	673	406	246	152	100	70	396

EC_1 method are observed, implying the quality improvement due to the employment of the proposed error compensation scheme. As shown in Fig. 10, the Horse sequence confirms a similar quality benefit of the proposed WFIP_1 and WFIP-EC_1 methods.

The second experiment is used to demonstrate the quality superiority and bitrate increase of the four variants of the proposed upsampling methods, the WFIP_1, WFIP_2, WFIP-EC_1, and WFIP-EC_2 methods, for different QPs when compared with the 6TF-based method, the WF-based method, the SAI method, and the IDID-EDI method. Tables I–VII show the resultant PSNR (peak signal-to-noise ratio) and bitrate of all concerned methods, and demonstrate the PSNR superiority of the two proposed upsampling methods with four variants with an acceptable bitrate increase. When compared with the 6TF-based method, the WF-based method, the SAI method, and the IDID-EDI method, the four proposed variants have average PSNR improvements ranging from 0.36 to 0.62 dB, 0.49 to 0.75 dB, 0.32 to 0.58 dB, and 0.32 to 0.58 dB, respectively. When the QP values are set to 16, 20, 24, 28, and 32, Tables I–VII indicate that the four proposed variants have better PSNR performance when compared with the four comparative methods; the average PSNR improvements range from 0.50 to 0.82 dB, 0.91 to 0.98 dB, 0.51 to 0.72 dB, and 0.48 to 0.80 dB, respectively. With the same QP, the proposed error compensation scheme is effective for the Rollerblade, Car, and Horse video sequences, but is less competitive for the other four sequences. The main reason is that the prediction capability of the proposed WFIP_1 and WFIP_2 methods is degraded due to heavy irregular textures, such as the lawn and leave regions, existing in the above three sequences, while the proposed WFIP-EC_1 and WFIP-EC_2 methods can efficiently compensate for the resultant estimation errors to achieve 0.13 dB and 0.21 dB PSNR improvements, respectively. When the QP values are set to 36 and 40, Tables I–VII illustrate the quality degradation of the four proposed variants when compared with that with lower QPs, and the PSNR improvement of the proposed methods is at most 0.2dB. In fact, ARSVC is rarely applied to the cases of high QPs because the serious quality degradation problem may cause visual discomfort in 3-D synthesized scenes. Consequently, the proposed upsampling methods do have sufficient quality benefit in practice, especially for low and middle QPs. Tables I–VII also illustrate that the IDID-EDI, WFIP_1,

TABLE III

QUALITY AND BITRATE COMPARISON FOR THE CAR VIDEO SEQUENCE FOR PSNR (DB) AND KILOBITS PER SECOND, RESPECTIVELY

Method \ QP		16	20	24	28	32	36	40	Average
6TF-based	PSNR	37.36	36.68	35.64	34.31	32.71	31.07	29.44	33.88
	Bitrate	1583	945	564	337	197	114	65	544
WF-based	PSNR	36.75	36.12	35.19	34.01	32.55	31.01	29.43	33.58
	Bitrate	1583	945	564	337	197	114	65	544
SAI	PSNR	37.16	36.56	35.44	34.00	32.52	30.91	29.38	33.71
	Bitrate	1583	945	564	337	197	114	65	544
IDID-EDI	PSNR	36.78	36.25	35.39	34.25	32.77	31.14	29.48	33.72
	Bitrate	1599	956	574	343	200	115	65	550
WFIP_1	PSNR	38.02	37.06	35.82	34.39	32.75	31.11	29.48	34.09
	Bitrate	1595	957	577	350	209	126	77	556
WFIP_2	PSNR	37.69	36.82	35.65	34.28	32.67	31.05	29.44	33.94
	Bitrate	1583	945	564	337	197	114	65	544
WFIP-EC_1	PSNR	38.21	37.22	35.94	34.47	32.79	31.12	29.48	34.17
	Bitrate	1607	969	588	361	220	137	89	567
WFIP-EC_2	PSNR	37.98	37.06	35.82	34.39	32.72	31.05	29.43	34.06
	Bitrate	1595	956	576	349	208	125	76	555

TABLE IV

QUALITY AND BITRATE COMPARISON FOR THE HORSE VIDEO SEQUENCE FOR PSNR (DB) AND KILOBITS PER SECOND, RESPECTIVELY

Method \ QP		16	20	24	28	32	36	40	Average
6TF-based	PSNR	30.89	30.75	30.44	29.82	28.53	26.78	25.34	28.94
	Bitrate	2135	1363	891	567	333	183	95	795
WF-based	PSNR	30.22	30.05	29.76	29.22	28.13	26.58	25.23	28.46
	Bitrate	2135	1363	891	567	333	183	95	795
SAI	PSNR	30.91	30.72	30.37	29.73	28.46	26.72	25.27	28.88
	Bitrate	2135	1363	891	567	333	183	95	795
IDID-EDI	PSNR	30.16	30.05	29.83	29.41	28.48	26.88	25.38	28.60
	Bitrate	2187	1400	916	583	342	186	96	816
WFIP_1	PSNR	31.49	31.25	30.79	29.99	28.56	26.77	25.33	29.17
	Bitrate	2147	1375	903	579	346	195	107	807
WFIP_2	PSNR	31.19	30.96	30.55	29.82	28.46	26.70	25.28	29.00
	Bitrate	2135	1363	891	567	333	183	95	795
WFIP-EC_1	PSNR	31.65	31.40	30.94	30.13	28.65	26.81	25.35	29.28
	Bitrate	2158	1387	915	591	357	207	119	819
WFIP-EC_2	PSNR	31.43	31.20	30.78	30.02	28.59	26.76	25.30	29.15
	Bitrate	2146	1375	902	578	345	194	107	807

TABLE V

QUALITY AND BITRATE COMPARISON FOR THE BULLINGER VIDEO SEQUENCE FOR PSNR (DB) AND KILOBITS PER SECOND, RESPECTIVELY

Method \ QP		16	20	24	28	32	36	40	Average
6TF-based	PSNR	38.08	37.67	36.95	35.96	34.44	32.69	30.88	35.24
	Bitrate	2588	1520	945	602	366	211	122	908
WF-based	PSNR	37.98	37.53	36.77	35.78	34.27	32.59	30.86	35.11
	Bitrate	2588	1520	945	602	366	211	122	908
SAI	PSNR	39.22	38.39	37.23	36.08	34.41	32.66	30.81	35.54
	Bitrate	2588	1520	945	602	366	211	122	908
IDID-EDI	PSNR	37.86	37.48	36.81	35.87	34.41	32.71	30.88	35.15
	Bitrate	2641	1564	974	632	389	220	124	935
WFIP_1	PSNR	40.03	39.17	37.96	36.57	34.71	32.80	30.96	36.03
	Bitrate	2600	1533	957	615	378	223	134	920
WFIP_2	PSNR	39.43	38.69	37.62	36.34	34.57	32.73	30.92	35.76
	Bitrate	2588	1520	945	602	366	211	122	908
WFIP-EC_1	PSNR	40.08	39.21	37.98	36.59	34.72	32.80	30.95	36.05
	Bitrate	2612	1544	969	626	389	234	145	931
WFIP-EC_2	PSNR	39.51	38.76	37.66	36.38	34.59	32.73	30.91	35.79
	Bitrate	2599	1532	956	614	377	222	133	919

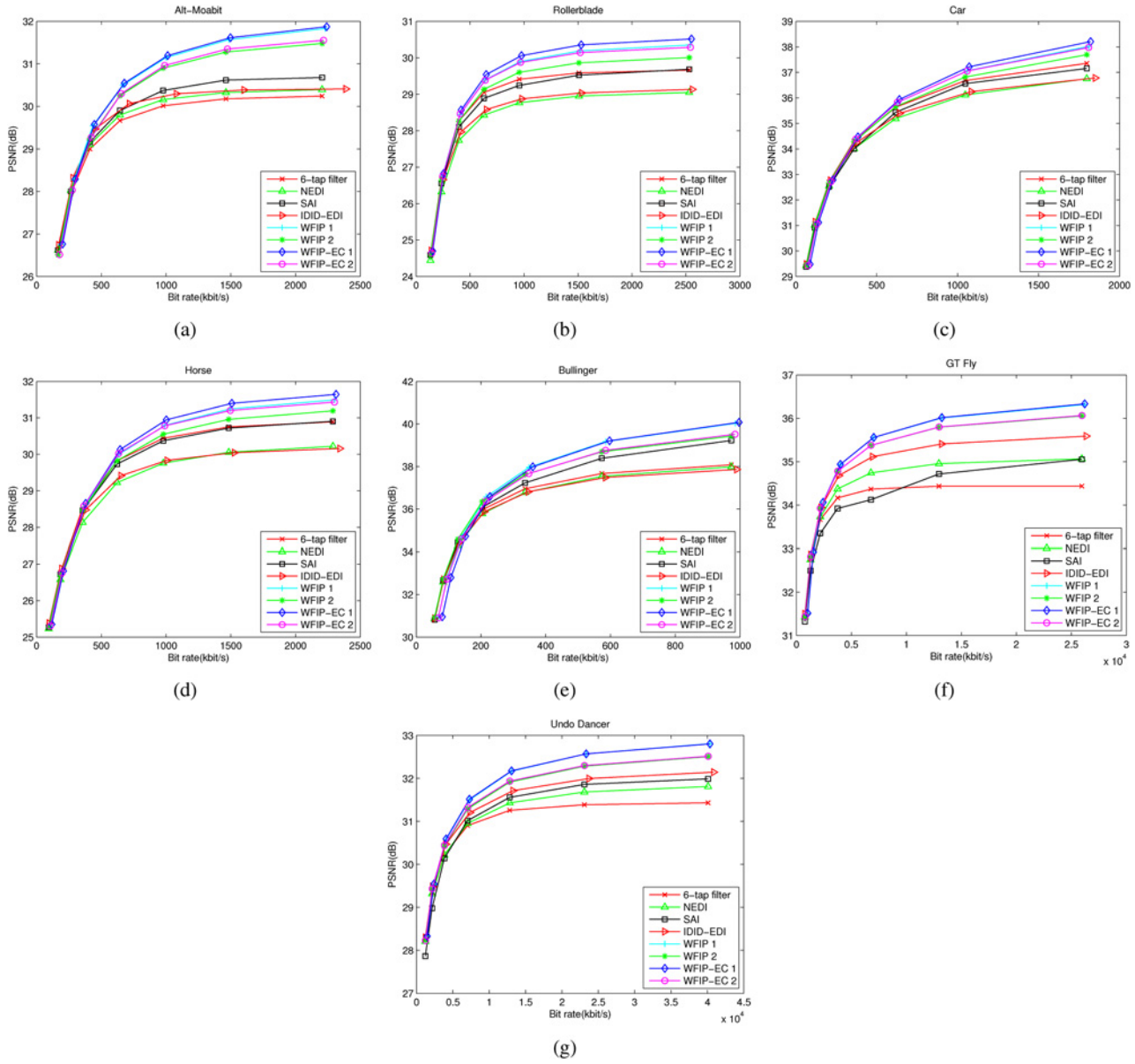


Fig. 11. RD curves of all concerned upsampling methods. (a) *Alt Moabit* sequence. (b) *Rollerblade* sequence. (c) *Car* sequence. (d) *Horse* sequence. (e) *Bullinger* sequence. (f) *GT Fly* sequence. (g) *Undo Dancer* sequence.

WFIP-EC_1, and WFIP-EC_2 methods result in a higher bitrate than the other four methods which do not cause any bitrate increase. We further provide Tables VIII-X to illustrate the bitrate increase in the proposed WFIP_1, WFIP-EC_1, and WFIP-EC_2 methods for different QPs. Due to an extra-bit bitstream requirement, the proposed WFIP_1 method presents an average bitrate increase of 2.35%–10.69%. Because it needs to save twelve coefficients for each right-view frame to compensate for the estimation error, the proposed WFIP-EC_1 and WFIP-EC_2 methods require a slightly higher bitrate than the proposed WFIP_1 and WFIP_2 methods, respectively, resulting in average bitrate increases of 5.32%–16.28% and 0.30%–7.89%, respectively.

According to the results of the second experiment, as shown in Fig. 11, the final experiment is used to demonstrate the rate distortion (RD) performance of all concerned methods. From Fig. 11, it is demonstrated that for smaller sized video sequences, that is, the Alt Moabit, Rollerblade, Car, Horse, and Bullinger sequences, and for a bitrate greater than 500 kbps, the four proposed variants have superior RD performance when compared with the previous four upsampling methods. On the contrary, due to the bitrate overhead caused by saving the twelve coefficients and/or extra-bit bitstream for error compensation, there is some decline in the RD performance of the four proposed variants, resulting in similar RD performance as the previous four upsampling methods, when the bitrate is less than 500 kbps. Although the RD performance of the four proposed variants is decayed for the low bitrate case, it seldom occurs in practice since encoding stereoscopic video sequences at a low bitrate, i.e. with high QP, leads to visual discomfort in 3-D synthesized scenes. From Fig. 11, for larger sized video sequences, such as the GT Fly and Undo

TABLE VI

QUALITY AND BITRATE COMPARISON FOR THE GT FLY VIDEO SEQUENCE FOR PSNR (DB) AND KILOBITS PER SECOND, RESPECTIVELY

Method \ QP		16	20	24	28	32	36	40	Average
6TF-based	PSNR	34.43	34.43	34.37	34.17	33.66	32.76	31.45	33.61
	Bitrate	20956	10661	5742	3307	1987	1195	719	6367
WF-based	PSNR	35.06	34.96	34.74	34.37	33.73	32.75	31.43	33.86
	Bitrate	20956	10661	5742	3307	1987	1195	719	6367
SAI	PSNR	35.05	34.71	34.12	33.92	33.35	32.50	31.33	33.57
	Bitrate	20956	10661	5742	3307	1987	1195	719	6367
IDID-EDI	PSNR	35.59	35.41	35.11	34.68	33.94	32.87	31.50	34.16
	Bitrate	20985	10686	5766	3329	2005	1208	726	6386
WFIP_1	PSNR	36.32	36.01	35.56	34.94	34.06	32.92	31.51	34.47
	Bitrate	20966	10670	5751	3316	1996	1204	728	6376
WFIP_2	PSNR	36.05	35.80	35.38	34.78	33.92	32.81	31.44	34.31
	Bitrate	20956	10661	5742	3307	1987	1195	719	6367
WFIP-EC_1	PSNR	36.33	36.02	35.56	34.94	34.06	32.92	31.52	34.48
	Bitrate	20977	10681	5762	3327	2007	1216	740	6387
WFIP-EC_2	PSNR	36.06	35.80	35.38	34.78	33.93	32.82	31.44	34.31
	Bitrate	20968	10672	5753	3318	1998	1207	731	6378

TABLE VII

QUALITY AND BITRATE COMPARISON FOR THE UNDO DANCER VIDEO SEQUENCE FOR PSNR (DB) AND KILOBITS PER SECOND, RESPECTIVELY

Method \ QP		16	20	24	28	32	36	40	Average
6TF-based	PSNR	31.43	31.39	31.26	30.90	30.23	29.34	28.24	30.40
	Bitrate	36108	20677	11671	6581	3653	2058	1146	11699
WF-based	PSNR	31.81	31.68	31.43	30.95	30.21	29.32	28.21	30.52
	Bitrate	36108	20677	11671	6581	3653	2058	1146	11699
SAI	PSNR	31.99	31.86	31.56	31.02	30.14	28.98	27.86	30.49
	Bitrate	36108	20677	11671	6581	3653	2058	1146	11699
IDID-EDI	PSNR	32.15	32.00	31.71	31.21	30.47	29.48	28.30	30.76
	Bitrate	36505	20916	11815	6697	3717	2079	1153	11840
WFIP_1	PSNR	32.81	32.57	32.17	31.50	30.58	29.54	28.33	31.07
	Bitrate	36353	20922	11916	6825	3898	2302	1390	11944
WFIP_2	PSNR	32.51	32.28	31.91	31.29	30.42	29.42	28.23	30.87
	Bitrate	36108	20677	11671	6581	3653	2058	1146	11699
WFIP-EC_1	PSNR	32.80	32.57	32.18	31.52	30.59	29.55	28.33	31.08
	Bitrate	36365	20933	11927	6837	3910	2314	1402	11955
WFIP-EC_2	PSNR	32.51	32.30	31.94	31.32	30.44	29.43	28.23	30.88
	Bitrate	36120	20689	11682	6592	3665	2069	1157	11711

TABLE VIII

BITRATE INCREASE RATIO OF THE PROPOSED WFIP_1 METHOD IN TERMS OF PERCENTAGE (%)

Sequence \ QP	16	20	24	28	32	36	40	Average
Alt_Moabit	1.04	1.58	2.36	3.58	5.59	8.72	13.86	5.25
Rollerblade	0.36	0.60	0.95	1.45	2.30	3.89	6.89	2.35
Car	0.68	1.17	1.99	3.40	5.94	10.43	18.92	6.07
Horse	0.53	0.82	1.25	1.97	3.44	6.64	13.00	3.95
Bullinger	1.26	2.13	3.64	5.99	9.51	14.64	21.53	8.39
GT Fly	0.94	1.89	3.61	6.53	11.27	19.00	31.61	10.69
Undo Dancer	0.61	1.06	1.91	3.49	6.40	11.39	20.25	6.44
Average	0.77	1.32	2.24	3.77	6.35	10.67	18.0	6.16

TABLE IX
BITRATE INCREASE RATIO OF THE PROPOSED WFIP-EC_1 METHOD IN TERMS OF PERCENTAGE (%)

Sequence \ QP	16	20	24	28	32	36	40	Average
Alt_Moabit	1.57	2.36	3.53	5.37	8.38	13.09	20.79	7.87
Rollerblade	0.82	1.37	2.16	3.28	5.21	8.80	15.60	5.32
Car	1.32	2.27	3.85	6.59	11.53	20.24	36.73	11.79
Horse	1.04	1.60	2.43	3.82	6.68	12.88	25.24	7.67
Bullinger	2.44	4.14	7.06	11.63	18.45	28.43	41.79	16.28
GT Fly	0.99	1.98	3.78	6.84	11.80	19.90	33.09	11.20
Undo Dancer	0.64	1.11	2.00	3.65	6.70	11.93	21.20	6.75
Average	1.26	2.12	3.54	5.88	9.82	16.47	27.78	9.53

TABLE X
BITRATE INCREASE RATIO OF THE PROPOSED WFIP-EC_2 METHOD IN TERMS OF PERCENTAGE (%)

Sequence \ QP	16	20	24	28	32	36	40	Average
Alt_Moabit	0.52	0.79	1.18	1.79	2.79	4.36	6.93	2.62
Rollerblade	0.46	0.76	1.21	1.83	2.91	4.91	8.71	2.97
Car	0.64	1.10	1.87	3.20	5.59	9.81	17.81	5.72
Horse	0.50	0.78	1.18	1.85	3.24	6.25	12.24	3.72
Bullinger	1.18	2.01	3.42	5.64	8.95	13.78	20.26	7.89
GT Fly	0.04	0.09	0.17	0.31	0.53	0.89	1.49	0.50
Undo Dancer	0.03	0.05	0.09	0.16	0.30	0.54	0.95	0.30
Average	0.48	0.79	1.30	2.11	3.47	5.79	9.7	3.39

Dancer sequences, the four proposed variants are superior to the previous four methods when the bitrate is greater than 5 Mbps.

The above three experiments have demonstrated that the four variants of our proposed upsampling methods can improve the quality of upsampled right-view frames for the ARSVC when compared with the four comparative methods, and the resultant bitrate overhead is negligibly small.

VI. CONCLUSION

This paper has presented the proposed WFIP and WFIP-EC upsampling methods for ARSVC. The proposed WFIP methods improve the traditional WF method by incorporating the information from the similarity between the left-view and right-view frames. By compensating for the prediction errors, the proposed WFIP-EC methods further improve the quality of the upsampled images obtained by the proposed WFIP methods. The proposed WFIP methods outperform, in terms of the RD curve, the existing methods in the cases of middle and high bitrate. In the case of low bitrate, the proposed WFIP methods suffer from the higher overhead proportion in the bitrate. The proposed WFIP-EC methods can further improve the performance of the proposed WFIP methods by compensating for the prediction errors, especially in the sequences with heavy irregular textures. The proposed methods achieve higher PSNR than the existing methods in the cases of middle and high bitrate. The proposed methods may achieve lower PSNR than the existing methods in the case of low bitrate, which seldom occurs in practice due to the visual discomfort in 3-D synthesized scenes. We conclude that the proposed upsampling methods are practical and improve the quality of the reconstructed video sequences for ARSVC.

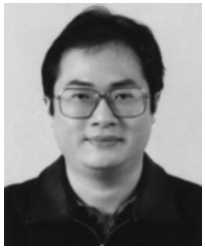
ACKNOWLEDGMENT

The authors appreciate the programming help of Mr. Y.-H. Shen and the proofreading help of Ms. C. Harrington and Prof. W.-N. Yang.

REFERENCES

[1] DVB Consortium. (2010, Jul.). *DVB commercial requirements for DVB 3-D-TV*, [Online]. Available: http://www.dvb.org
[2] A. Smolic, K. Mueller, N. Stefanoski, J. Ostermann, A. Gotchev, G.B. Akar, G. Triantafyllidis, and A. Koz, "Coding algorithms for 3-D-TV - A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1606–621, Nov. 2007.
[3] A. Aksay and G. B. Akar, "Evaluation of stereo video coding schemes for mobile devices," in *Proc. True Vision Capture, Transmission Display 3-D Video*, May 2009, pp. 1–4.
[4] P. Seunti'ns, L. Meesters, and W. IJsselsteijn, "Perceived quality of compressed stereoscopic images: Effects of symmetric and asymmetric JPEG coding and camera separation," *ACM Trans. Appl. Percept.*, vol. 3, no. 2, pp. 95–109, Apr. 2006.
[5] C. Fehn, P. Kauff, S. Cho, H. Kwon, N. Hur, and J. Kim, "Asymmetric coding of stereoscopic video for transmission over T-DMB," in *Proc. 3-D-TV Conf.*, May 2007, pp. 1–4.
[6] Y. Chen, Y. K. Wang, M. M. Hannuksela, and M. Gabbouj, "Picture-level adaptive filter for asymmetric stereoscopic video," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 1944–1947.
[7] Y. Chpten, Y. K. Wang, M. Gabbouj, and M. M. Hannuksela, "Regionally Adaptive Filtering for Asymmetric Stereoscopic Video Coding," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2009, pp. 2585–2588.
[8] S. N. Park and D. G. Sim, "View-dependency video coding for asymmetric resolution stereoscopic views," *Opt. Eng.*, vol. 48, no. 7, pp. 1–8, Jul. 2009.
[9] P. Aflakt, M. M. Hannuksela, J. Hakala, J. Hakkinen, and M. Gabbouj, "Joint adaptation of spatial resolution and sample value quantization for asymmetric stereoscopic video compression: A subjective study," in *Proc. Int. Symp. Image Signal Process. Anal.,* Feb. 2012, pp. 396–401.
[10] M. Yu, H. Yang, S. Fu, F. Li, and R. Fu, "New sampling strategy in asymmetric stereoscopic video coding for mobile devices," in *Proc. Int Conf. E-Product E-Service E-Entertainment,* Dec. 2010, pp. 1–4.

[11] P. Aflakia, M. M. Hannuksela,M. Homayouni, and M. Gabbouj, "Cross-asymmetric mixed-resolution 3d video compression," in *Proc. 3-D-TV-Conf.: True Vision—Capture, Transmission Display 3-D Video,* Oct. 2012, pp. 1–4.
[12] F. Shao, G. Jiang, M. Yu, K. Chen, and Y. S. Ho, "Stereoscopic video coding with asymmetric luminance and chrominance qualities," *IEEE Trans. Consum. Electron.,* vol. 56, no. 4, pp. 2460–2468, Nov. 2010.
[13] F. Shao, G. Jiang, M. Yu, and H. Dong, "JND-based asymmetric coding of stereoscopic video for mobile 3-D-TV applications," in *Proc. Int. Cong. Image Signal Process.,* Oct. 2011, pp. 186–189.
[14] F. Shao, G. Jiang, M. Yu, and Y. S. Ho, "Asymmetric coding of multiview video plus depth based 3-D video for view rendering," *IEEE Trans. Multimedia,* vol. 14, no. 1, pp. 157–167, Feb. 2012.
[15] P. Aflaki, M. M. Hannuksela, H. Sarbolandi, and M. Gabbouj, "Simultaneous 2D and 3-D perception for stereoscopic displays based on polarized or active shutter glasses," *J. Visual Commun. Image Represent.,* to be published.
[16] T. Na, S. Ahn, H. Sabirin, M. Kim, B. Kim, S. Hahm, and K. Lee, "A hybrid stereoscopic video coding scheme based on MPEG-2 and HEVC for 3-D-TV Services," *IEEE Trans. Circuits Syst. Video Technol.,* to be published.
[17] W. Yao, L. P. Chau, and S. Rahardja, "Joint rate allocation of stereoscopic 3d videos in next- generation broadcast applications," *IEEE Trans. Broadcast.* to be published.
[18] P. Aflaki, M. M. Hannuksela, J. Hakkinen, P. Lindroos, and M. Gabbouj, "Subjective study on compressed asymmetric stereoscopic video," in *Proc. IEEE Int. Conf. Image Process.,* Sep. 2010, pp. 4021–4024.
[19] G. Saygili, C. G. Gurler, and A. M. Tekalp, "Evaluation of asymmetric stereo video coding and rate scaling for adaptive 3-D video streaming," *IEEE Trans. Broadcast.,* vol. 57, no. 2, pp. 593–601, Jun. 2011
[20] ISO/IEC JTC1/SC29/WG11 and ITU-T SG16/Q.6, Adaptive Basic Unit Layer Rate Control for JVT, Doc. JVT-G012. Pattaya, Thailand, 2003.
[21] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.,* vol. 10, no. 10, pp. 1521–1527, Oct. 2001.
[22] X. Zhang, "Image interpolation by adaptive 2-D autoregressive modeling and soft-decision estimation," *IEEE Trans. Image Process.,* vol. 17, no. 6, pp. 887–896, Jun. 2008.
[23] Y. Zhang, D. Zhao, J. Zhang, R. Xiong, and W. Gao, "Interpolation-dependent image downsampling," *IEEE Trans. Image Process.,* vol. 20, no. 11, pp. 3291–3296, Nov. 2011.
[24] M. G. Perkins, "Data compression of stereopairs," *IEEE Trans. Commun.,* vol. 40, no. 4, pp. 684–696, Apr. 1992.
[25] ISO/IEC MPEG and ITU-T VCEG, Draft Reference Software for MVC, Doc. JVT-AE207. London, U.K., 2009.
[26] MOBILE3DTV (2009, Feb.). *Mobile 3-D-TV content delivery optimization over DVBH system* [Online]. Available: http://sp.cs.tut.fi/mobile3dtv/stereo-video



Kuo-Liang Chung (M'91–SM'01) received the B.S., M.S., and Ph.D. degrees from National Taiwan University, Taipei, Taiwan, in 1982, 1984, and 1990, respectively.

He is currently a Professor with the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan, where he has been since 1995. After two years of obligatory military service from 1984 to 1986, he was a Research Assistant with the Institute of Information Science at Academia Sinica, Taipei, Taiwan, from 1986 to 1987. He was a Visiting Scholar with the University of Washington, Seattle, USA, in 1999. From 2003 to 2006, he was the Chair of the Department of Computer Science and Information Engineering, NTUST. Since 2009, he has been a University Chair Professor with NTUST. His current research interests include image or video compression, image or video processing, pattern recognition, 3-D video processing, and shape analysis in computer vision.

Dr. Chung was a recipient of the Distinguished Research Award from the National Science Council of Taiwan in 2004, the Best Paper Award from the Image Processing and Pattern Recognition Society of Taiwan in 2007, and the Distinguished Teaching Award of NTUST in 2009. He had been a Managing Editor of the *Journal of Chinese Institute of Engineers* from 1996 to 1998. In August 2000, he was the Program Co-Chair of the Conference on Computer Vision, Graphics, and Image Processing, Taiwan. He is currently an Associate Editor of the *Journal of Visual Communication and Image Representation*. He is a fellow of the Institute of Engineering and Technology.

Yong-Huai Huang (M'11) received the B.S. degree in information management from Aletheia University, Danshui, Taipei, Taiwan, and the M.S. and Ph.D. degrees in computer science and information engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan.

He is currently an Associate Professor in the Department of Electronic Engineering, Jinwen University of Science and Technology, New Taipei City, Taiwan. His current research interests include image processing and compression, and algorithms.

Wen-Chang Liu received the B.S. degree in computer science and information engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan.

His current research interests include image processing and compression, and algorithms.