A Low Complexity Compressed Sensing-Based Codec for Consumer Depth Video Sensors

Shengwei Wang, Li Yu, *Member, IEEE*, and Sen Xiang, *Member, IEEE*

***Abstract*—In 3-D applications, a high-quality depth video codec with low complexity is in great demand for consumer devices, which generally have limited computational resources and energy. Based on the compressed sensing theory, we propose a low complexity depth video codec to compress depth videos effectively. The proposed codec decomposes depth blocks via an adaptive wavelet decomposition algorithm by following the principle of local entropy minimization. The decomposition can reduce the amount of local data, and separate depth blocks by frequency precisely. In order to reduce temporal redundancy, a block average value-based fast motion estimation scheme is also designed. Further, a joint optimization method is proposed to select the best combination of quantization parameter and measurement rate for each block. The experimental results demonstrate that the proposed codec, compared with H.265 and H.264, achieves BD-PSNR improvement up to 1.34 dB and 4.28 dB at most respectively in "AllIntra" mode. In "IPPP" mode, the proposed codec also achieves BD-PSNR improvement up to 0.62 dB and 1.79 dB on average respectively. Moreover, compared with other methods, the complexity and energy consumption of the proposed codec is much lower, which is adapted to consumer devices well.**

***Index Terms*—Compressed sensing, depth video, measurement rate, optimization.**

I. Introduction

Depth image based rendering (DIBR) [1] makes it possible to render virtual views from texture and depth (T+D) images of reference views. Based on DIBR, three dimension (3D) applications have shown great potential in consumer electronics. In T+D videos, compression of depth videos play an important role. Particularly, depth video sensors in consumer devices such as cellphones, drones, are required to implement depth video compression with low complexity. Moreover, virtual reality (VR) applications make it possible for users to capture, view and disseminate whole real scenes freely with consumer terminals. Above applications have shown high promises, while they all require a depth video codec with low-complexity and energy-saving to compress depth videos.

For depth sensors of consumer devices, coding complexity should be considered carefully, since these devices generally have limited computational resources and energy. Existing coding standards such as H.264/AVC [2], H.265/HEVC [3] cannot meet the demand of depth video sensors due to their high-complexity. Recently, the compressed sensing theory (CS) [4] provides a new method to compress videos with low-complexity. The CS theory can sample signals by sample rates below the Nyquist sampling theorem according to sparsity of signals, which can be used to compress depth videos efficiently. Additionally, the sampling process is simple, which can reduce the complexity and save power.

Based on the CS theory, we propose a novel depth video codec with low-complexity for consumer devices. In the proposed codec, the encoder processes frames according to the type of frames. For I frames, the encoder applies an adaptive wavelet transform to decompose blocks, following the principle of local entropy minimization. The wavelet transform separates high and low frequencies, making it convenient to sample different frequency bands in different measurement rates. For P frames, a block average value based fast motion estimation scheme is designed to reduce temporal redundancy in successive frames. During the process of encoding, a joint optimization method is developed to select the best combination of quantization parameter (QP) and measurement rate for each block, in order to achieve better coding performance.

The remainder of the paper is organized as follows: Section II introduces related works. In Section III, the compressed sensing theory is explained. Section IV shows the proposed codec. Section V explains the joint optimization of QP and measurement rate. The experiments are conducted in Section VI. Finally, Section VII draws the conclusion.

II. Related Work

Aiming at reducing the complexity of traditional video coding standards, many methods [5], [6], [7] have been proposed recently. In order to reduce the complexity of motion estimation, an algorithm, which consists of sub-sampling, data reuse, pixel truncation and adaptive search range, was proposed in [6]. Singhadia *et al.* [7] proposed a memory efficient addressing algorithm for DCT in HEVC, which has a low power dissipation. Meanwhile, Garrido *et al.* [8] designed an architecture to implement the multiple transform of VVC [9] on FPGA chips with a moderate consumption of hardware

Manuscript received September 14, 2018; revised January 31, 2019, April 4, 2019, June 18, 2019, and July 12, 2019; accepted July 15, 2019. Date of publication July 18, 2019; date of current version October 24, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61871437, in part by the Natural Science Foundation of Hubei Province under Grant 2017CFB348, in part by the Research Foundation of Hubei Educational Committee under Grant Q20171106, and in part by the Research Foundation for Young Scholars of WUST under Grant 2017xz008. *(Corresponding author: Li Yu.)*

S. Wang and L. Yu are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: kadinwang@hust.edu.cn; hustlyu@hust.edu.cn).

S. Xiang is with the School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081, China (e-mail: xiangsen@wust.edu.cn).

Digital Object Identifier 10.1109/TCE.2019.2929586

resources. Wang *et al.* [10] designed a parallel loop filtering scheme on GPU for HEVC. Moreover, Kim and Lee [11] segmented input videos into the foreground and background to control the complexity respectively, achieving low-power surveillance video coding.

Compared with traditional standard encoders, the CS theory provides a novel way to encode depth videos. By utilizing the low sampling rate and the simple process of sampling, researchers have developed many CS-based video encoders [12], [13]. A CS-based video encoder mainly consists of transforming, quantization, sampling, and entropy coding.

In depth video encoders, transforming converts images from spatial domain to frequency domain. An existing CS-based depth map coding algorithm [14] adopts a sub-sampled 2D discrete cosine transform (DCT) to obtain de-correlated samples. Traditional DCT is not efficient for maintaining complex shaped edges [15] which are important for depth maps. Graph based transform (GBT) has also been proposed for depth video coding [16], providing effective sparse transform. Nevertheless, side information of GBT reduces the overall coding efficiency. By applying Fourier transform, Sarkis and Diepold [17] proposed a depth coding algorithm with the variable density random sampling method. The method does not consider the characteristics of Fourier coefficients which can be compressed. While Li *et al.* [18] utilized Gaussian mixture models (GMM) to generate transformed product vector quantizers, the number of Gaussian components limits the selection of bit rates. Considering that depth maps contain both sharp boundaries and smooth areas, an adaptive wavelet transform is developed in the proposed codec. By transforming, sharp boundaries are concentrated in high frequency sub-blocks, while large smooth areas lay in low frequency sub-blocks. Furthermore, the transform follows the principle of local entropy minimization, reducing the local data volume.

Many existing inter prediction methods cannot be applied in CS-based encoders for their high complexity. To reduce temporal redundancy, Vijayanagar *et al.* [19] directly compressed the residual information between blocks in P frames and the co-located blocks in I frames, ignoring the motion information between successive frames. Similarly, Li *et al.* [18] compressed temporal redundancy between adjacent frames by using the differential symbol encoding method. Chen *et al.* [20] exploited temporal correlation among frames. In his method, a frame level motion estimation is conducted to reduce temporal redundancy. Nevertheless, the process of motion estimation can be more precise and applied in block level. Liu *et al.* [21] developed a block level motion estimation scheme in the distributed codec. In the method, motion estimation is conducted in the decoder, and data is inversely transmitted to the encoder. The additional backward data transmission increases communication burden. Based on the extracted average value of the block, in the proposed codec, a block-level fast motion estimation is designed to reduce the temporal redundancy with negligible complexity.

In CS-based depth video encoder, the proper combination of QP and measurement rate optimizes bit-rate and coding quality. Liu *et al.* [22] predefined QP and measurement rate empirically, which was lack of flexibility. Do *et al.* [23] employed a fixed measurement rate, and adjusted QP to control the coding process, which cannot reach the best performance. In some methods, such as [20], [24], dictionaries are trained as references to measurement rates of frames. Although this kind of methods is highly efficient, the measurement rate relays on the quality of dictionaries heavily. Cen *et al.* [13], [25] deduced a rate-distortion (R-D) model with packet loss probability to guide the measurement rate allocation. Li *et al.* [12] proposed an adaptive cluster sparsity sensing method and dynamically determined the measurement rate. Compared with the above methods, the influence of QP and measurement rate on bit-rate and distortion is comprehensively considered in the proposed codec. By constructing a combined R-D model, the encoder can simultaneously determine the optimized QP and measurement rate to improve overall R-D performance.

III. Compressed Sensing in Depth Video Coding

This section introduces the process of compressed sensing and reconstruction, especially in the proposed depth codec. Consider an $N \times N$ block in depth videos, and vectorize the block as $\mathbf{x} \in \mathbf{R}^{N^2}$ by zig-zag scanning. By adopting a designed orthogonal basis $\Psi \in \mathbf{R}^{N^2 \times N^2}$, the block signal $\mathbf{x}$ can be represented as

$$\mathbf{x} = \Psi \mathbf{s} \tag{1}$$

where vector $\mathbf{s} \in \mathbf{R}^{N^2}$ is the sparse representation of $\mathbf{x}$. If $\mathbf{s}$ has K non-zero elements, $\mathbf{x}$ is a K-sparse signal [26].

In the CS-based depth codec, sampling is conducted by a linear measurement matrix $\Phi \in \mathbf{R}^{M \times N^2}$ with $M \ll N^2$ as

$$\mathbf{y} = \Phi \mathbf{x} = \Phi \Psi \mathbf{s} \tag{2}$$

where $\mathbf{y} \in \mathbf{R}^M$ is the sampled and compressed vector.

To reconstruct depth blocks, total variation (TV) minimization [27] is adopted in the proposed codec. TV minimization utilizes piece-wise smooth characteristics of depth blocks. Instead of finding the sparsest solution within the domain of transform Ψ, TV minimization finds the 'smoothest' solution within the space of possible solutions. TV minimization enforces sparsity upon the gradient of the solution. In detail, for a block B, TV function is defined as

$$TV(B) = \sum_{i,j} \left|B_{i+1,j} - B_{i,j}\right| + \left|B_{i,j+1} - B_{i,j}\right| \tag{3}$$

where $B_{i,j}$ represents the pixel value at position (i, j) in B. Applying the above function, the CS recovery problem can be solved by TV minimization as

$$\hat{\mathbf{x}} = \arg\min_{\hat{\mathbf{x}}} \left\|\mathbf{y} - \Phi\hat{\mathbf{x}}\right\|_2 + \lambda TV(\mathbf{x}) \tag{4}$$

where $\hat{\mathbf{x}}$ is the reconstructed vectorized depth block.

IV. Proposed Depth Codec Architecture

The proposed encoder is shown in Fig. 1. In the encoder, I frame is divided into the non-overlap 64×64 block, which is depicted as B_I. The average value of B_I is firstly calculated and transmitted via differential encoding, and the average value is depicted as m. After extracting m, B_I leaves residual values

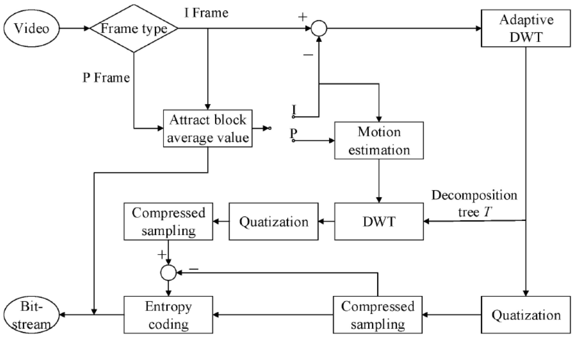


Fig. 1. The proposed depth video encoder.

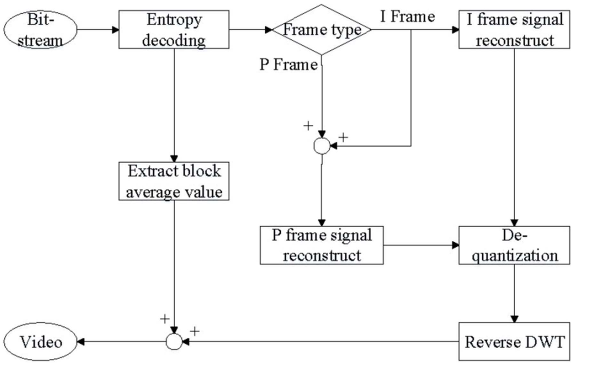


Fig. 2. The proposed depth video decoder.

as $B_{I,res}$. Then, an adaptive DWT is applied on $B_{I,res}$. The details of the adaptive DWT is presented in Section IV-A. Then, sub-blocks decomposed by the adaptive DWT are vectorized, quantized and compressive sampled as $\mathbf{y}_I$, which is explained in Section IV-C. To further reduce the data volume, a 0-order exp-Golomb code is adopted as the entropy coder to compress the sampling signal $\mathbf{y}_I$.

The encoding process of P frames is similar to I frames. Firstly, the block average value m is also calculated and transmitted. Different from I frames, the encoder applies fast motion estimation scheme for P frames to remove temporal redundancy. The motion estimation scheme utilizes m to search the best matching block $B_{I,ref}$ in the reference I frame efficiently, which is shown in Section IV-B. After motion estimation, the P frame is decomposed by DWT, following the decomposition tree T from $B_{I,ref}$. Next, the wavelet sub-block is also vectorized, quantized and compressed as $\mathbf{y}_P$ with the QP and the measurement rate from $B_{I,ref}$. Hence, the encoder calculates the residual of the sample signal $\mathbf{y}_P$ as

$$\mathbf{y}_{P,res} = \mathbf{y}_P - \mathbf{y}_I \tag{5}$$

where $\mathbf{y}_{P,res}$ is the residual of $\mathbf{y}_P$. Finally, $\mathbf{y}_{P,res}$ is converted into the bit-stream via entropy coding.

The decoder is presented in Fig. 2. By entropy decoding, the block average value m, the signal $\mathbf{y}_I$ and $\mathbf{y}_{P,res}$ are obtained. For I frames, the original wavelet sub-block can be reconstructed via TV minimization from $\mathbf{y}_I$ by (3) (4). Next, the residual block $B_{I,res}$ can be obtained by de-quantization and reverse DWT. After adding the extracted average value m, the original block B_I can be recovered. For P frames, the obtained residual signal $\mathbf{y}_{P,res}$ should be recovered by

$$\mathbf{y}_P = \mathbf{y}_{P,res} + \mathbf{y}_I \tag{6}$$

where $\mathbf{y}_I$ is the sample signal form $B_{I,ref}$. After recovering $\mathbf{y}_P$, the remaining process is the same as I frames.

A. Adaptive Wavelet Decomposition

In order to decompose $B_{I,res}$ efficiently, we designs an adaptive DWT decomposition scheme according to the principle of local entropy minimization. In the scheme, the block can be maximally decomposed to 8 × 8 sub-blocks via 3-level DWT if necessary. Considering that different blocks have different local entropy and edge information, the depth of decomposition is controlled by the principle of local entropy minimization. By utilizing the principle, the scheme can reduce the total local entropy of the whole residual block, and improve the sparsity of sub-blocks, which can lower the measurement rate in the following CS sampling. Meanwhile, the process of decomposition is lossless, which cannot lead to the quality degradation of residual blocks.

To be specific, 64 × 64 residual block $B_{I,res}$ is firstly decomposed to 32 × 32 sub-blocks B_i $(i = 1, 2, 3, 4)$ with a decomposition tree T constructed simultaneously. In the following decomposition, the local entropy is introduced as the cost function to control the decomposition process. For a block B, the local entropy is calculated as

$$H(B) = E\big[-\log p_i\big] = -\sum_{i=1}^{n} p_i \log p_i \tag{7}$$

where n is the total number of pixel values in the block, and p_i is the probability of occurrence of the i-th pixel value.

Considering that each block B_i can generate 4 sub-blocks via DWT, the total local entropy of 4 sub-blocks $SumH$ can be calculated, and is used to compare with the original block entropy $H(B_i)$. If $H(B_i)$ is larger, indicating that the decomposition reduces the total local entropy, 4 sub-blocks are reserved and T is updated. Otherwise, the decomposition for B_i is terminated and 4 sub-blocks are abandoned, in order to avoid increasing the local entropy. When the size of sub-block becomes 8 × 8 or all of the decomposition is terminated, the algorithm outputs all of the sub-blocks and the updated T.

Same as depth blocks, residual blocks also consist of sharp boundaries and smooth regions. After the adaptive DWT, boundaries and regions can be precisely transformed to high and low frequency sub-blocks respectively. Thus, different QPs and measurement rates can be selected for different sub-blocks to satisfy the property and the error-tolerance of each sub-block. Moreover, the wavelet basis is also an orthonormal sparse basis, lying the foundation for CS sampling.

B. Average Value Based Motion Estimation

In order to compress temporal redundancy of frames, a block average value based motion estimation scheme is designed for P frames. The scheme takes I frame in each GOP as the reference frame to predict the remaining P frames in the same GOP. Since similar blocks have similar average values,

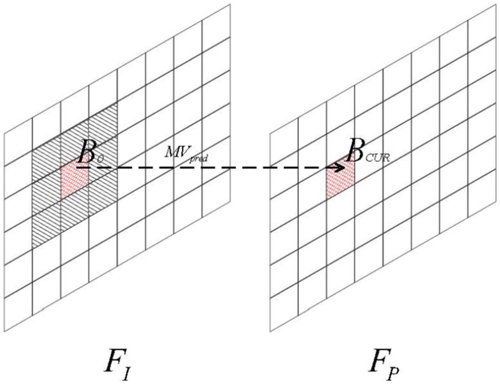


Fig. 3. Average value based fast motion estimation.

the scheme searches the best matching block in the reference I frame by comparing the block average value m. The average value based scheme can find the best candidate reference blocks for motion estimation. Meanwhile, the scheme has low complexity, reducing coding time and energy.

In detail, the designed motion estimation scheme is shown in Fig. 3. The red block in P frame F_P is the current coding block, and denoted as B_{Cur}. The average value of B_{Cur} is m_{Cur}. In order to predict B_{Cur}, the scheme selects the co-located block B_0 with the adjacent 8 blocks B_i $(i = 1, 2 \cdots 8)$ in the reference I frame F_I as the candidate blocks. Then, the average value m_i $(i = 1, 2 \cdots 8)$ of the candidate block B_i is compared with m_{Cur} respectively. If $|m_{Cur} - m_i| \leq 1$, the i-th block B_i is chosen as the candidate matching block. Among all the candidate matching blocks, the block with the least sum of absolute differences (SAD) is finally chosen as the reference block. SAD of block B_i is calculated as

$$SAD_i = \sum_{x=1}^{64} \sum_{y=1}^{64} |B_i(x, y) - B_{Cur}(x, y)| \tag{8}$$

where $B_i(x, y)$ and $B_{Cur}(x, y)$ are the pixel values of block B_i and B_{Cur} at position (x, y) respectively.

While none of the candidate blocks satisfies $|m_{Cur} - m_i| \leq 1$, the block which has the closest average value m_i to m_{Cur} is selected as the reference block directly. After the reference block is determined, the current block B_{Cur} is decomposed via DWT, following the decomposition tree T obtained from the reference block. In the following coding process, QP and measurement rate is also obtained from the reference block.

C. Quantization and Sampling

After the adaptive DWT, the obtained wavelet sub-blocks are quantized for better sampling. Considering that high frequency represents boundaries and plays a more important role in depth videos, QP is respectively set for high and low frequency sub-blocks. In the proposed codec, QP of high frequency sub-blocks is set as $Q_H = k$ $(k = 0.1, 0.2, 0.3 \cdots)$, while QP of low frequency sub-blocks is $Q_L = 5k$. After determining QP, the quantized values of the wavelet sub-block is shown as

$$s_q(x, y) = round\left(\frac{s(x, y)}{Q}\right) \tag{9}$$

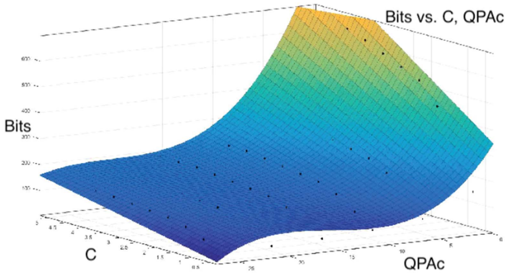


Fig. 4. The surface of 'Bits' to (C, Q_H). Notes: 'Bits' means total bits needed to encode blocks, 'C' and 'QPAc' indicate the measurement coefficient C and Q_H respectively.

for simplification, the subscript L and H are omitted. Where s is the original value, and s_q is the quantized value. (x, y) is the position of the value.

After quantization, the quantized sub-block is vectorized via zig-zag scanning as vector $\mathbf{s}_q$. Then, a partial Hadamard matrix [28] is adopted to sample $\mathbf{s}_q$ into $\mathbf{y}$ by (2).

V. Combined R-D Model and Joint Optimization

In the proposed codec, QP and measurement rate both directly affect the coding performance. Thus, a joint optimization method of QP and measurement rate is proposed to improve the coding quality. In order to determine the best combination of QP and measurement rate, the influences of above two parameters to bit-rate and distortion are studied respectively. Due to $Q_L = 5Q_H$, Q_H is used to indicate QP of blocks.

The measurement rate is defined as

$$MeasurementRate = \frac{M}{N} \tag{10}$$

where M is the number of elements of the sampling signal, and N is the number of elements of the original vectorized signal. Further, M can be calculated as

$$M = CK \log \frac{N}{K} \tag{11}$$

where K is the sparsity degree, and C is the undetermined coefficient. It can be seen that the measurement rate is controlled by C from (10) and (11). Therefore, we select C to indicate the measurement rate.

The relationship between the bits needed to encode blocks and the combined (C, Q_H) is studied. By statical results, we find that various depth sequences have similar relationship, and show the same properties. As an example, Fig. 4 shows the surface of Bits to (C, Q_H) of the first frame from 'Undo Dancer'. In the figure, 'Bits' means total bits needed to encode blocks, 'C' and 'QPAc' indicate the measurement coefficient C and Q_H respectively. In the surface, the bits increase with the growth of C, and drop when Q_H increases. Then, considering complexity and performance, a polynomial function can be used to fit the surface as

$$Bits = p_{02}Q_H^2 + p_{11}Q_H C + p_{01}Q_H + p_{10}C + p_{00} \tag{12}$$

where $p_{02}, p_{11}, p_{01}, p_{10}$ and p_{00} are undetermined coefficients.

Same as the above analysis, the surface of block distortion to (C, Q_H) can be obtained as Fig. 5 shows. The mean squared error (MSE) is used to evaluate the block distortion.

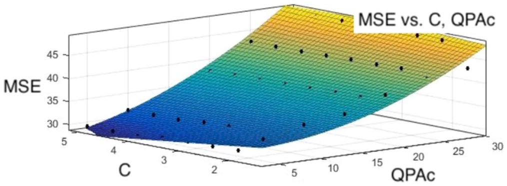


Fig. 5. The surface of 'MSE' to (C, Q_H). Notes: 'MSE' means distortion of blocks, 'C' and 'QPAc' indicate the measurement coefficient C and Q_H respectively.

TABLE I
R^2 COEFFICIENT OF FITTING FUNCTION

Sequences	Bits Function	MSE Function
Poznan_Hall2	0.9237	0.9658
Undo_Dancer	0.9448	0.9765
Kendo	0.9362	0.9792
Newspaper	0.9385	0.9697
GT_Fly	0.9287	0.9716
Balloons	0.9408	0.9731

'C' and 'QPAc' indicate the measurement coefficient C and Q_H respectively. The fitting function is

$$MSE = q_{02}Q_H^2 + q_{11}Q_H C + q_{01}Q_H + q_{10}C + q_{00} \quad (13)$$

where q_{02}, q_{11}, q_{01}, q_{10} and q_{00} are undetermined coefficients.

For the sequences used for statistic, R^2 coefficients are listed in Table I. From the table, it is obvious to see that the proposed polynomial functions fit the actual surface well.

After obtaining (12) and (13), the QP and the measurement rate optimization problem can be transformed as

$$\begin{aligned} \min \ & MSE = f(C, Q_H) \\ s.t. \ & Bits = g(C, Q_H) \le Bits_T \end{aligned} \quad (14)$$

where $f(C, Q_H)$ indicates (13), $g(C, Q_H)$ indicates (12), and $Bits_T$ is the target bits to encode blocks.

In order to solve the problem, the problem is converted into an unconstrained problem as

$$\mathop{arg\min}_{C,Q_H} J = f(C, Q_H) + \lambda(g(C, Q_H) - Bits_T) \quad (15)$$

Gradient descent method is applied to solve the above problem. In detail, the initial solution of (15) is set as

$$Opt_0 = \begin{bmatrix} \lambda_0 \\ C_0 \\ Q_{H,0} \end{bmatrix} \quad (16)$$

Then, the gradient of (15) can be calculated as

$$\nabla J = \partial J \begin{bmatrix} \frac{1}{\partial \lambda} \\ \frac{1}{\partial C} \\ \frac{1}{\partial Q_H} \end{bmatrix} \quad (17)$$

Next, the iteration begins as

$$Opt_{k+1} = Opt_k - \alpha \nabla J \quad (18)$$

Until $\nabla J(Opt_k) \le \varepsilon$, the optimization Opt^* is set as Opt_k. Opt^* is finally used to encode the block.

According to (12) and (13), pre-encoding should be conducted at least 5 times, in order to generate fitting functions. In practice, every block in I frames is pre-encoded once, in order to reduce complexity. In detail, the current block can get encoded data from adjacent blocks, considering that adjacent blocks have spatial redundancy and similar properties with the current block. The data from left, upper, upper left and upper right encoded blocks can in total provide 8 groups of data to the current block, including 4 groups of pre-encoding data and 4 groups of coded data. Adding the pre-encoding data of the current block, 9 groups of data can be used to derive the fitting function, and calculate the optimized combination of QP and measurement rate. For P frames, the coding parameter is directly obtained from the reference block in the I frame.

TABLE II
EXPERIMENTAL SEQUENCES

Sequences	Left View	Right View	Virtual View
Poznan_Hall2	View 6	View 7	View 6.5
Poznan_Street	View 4	View 5	View 4.5
Undo_Dancer	View 1	View 5	View 3
GT_Fly	View 5	View 9	View 7
Kendo	View 1	View 3	View 2
Balloons	View 1	View 3	View 2
Newspaper	View 2	View 4	View 3
Shark	View 1	View 5	View 3

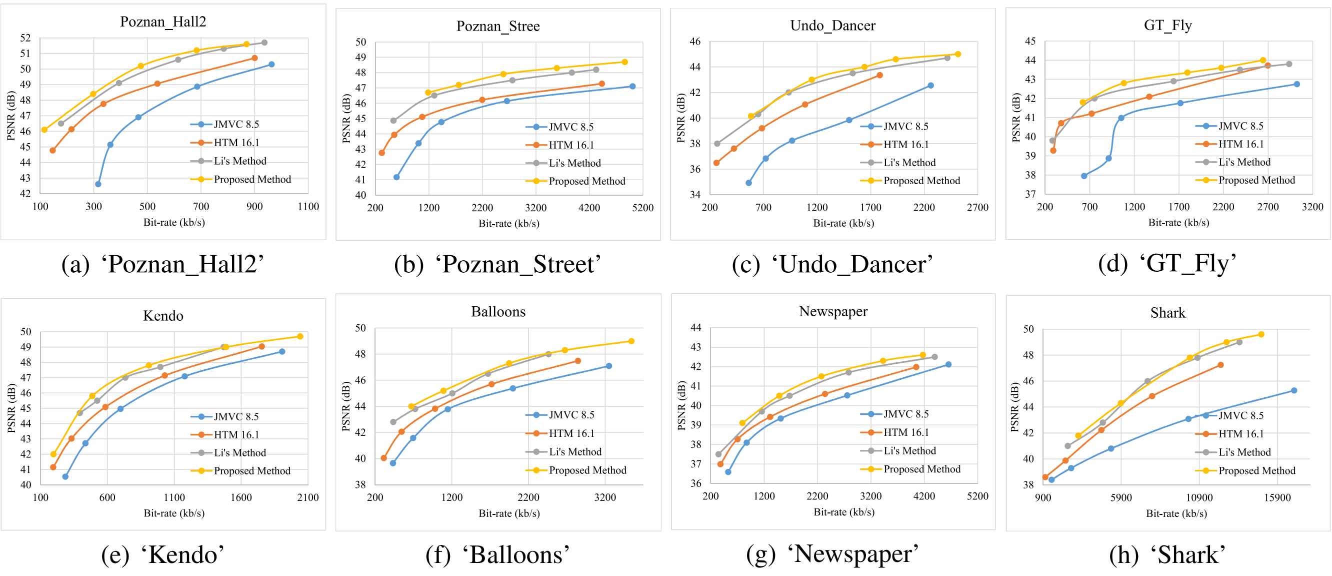
VI. EXPERIMENTAL RESULTS

Since depth videos are used to render virtual views in practice, virtual view videos are adopted to evaluate the performance of the proposed depth video codec. In detail, the experiments are conducted on 8 sequences, and 2 views of each sequence are selected as reference views to render a virtual view as Table II shows. All of the sequences and views are selected according to the common test condition of 3D video coding [29] from joint collaborative team on video coding (JCT-VC) which is the video standard working group. Generally speaking, a depth video codec should satisfy practical applications, if the codec could achieve high performance on these test sequences and views with diversified properties.

In order to verify the performance of the proposed codec on consumer electronics, all of the experiments are conducted on a cellphone. Since cellphones are widely used in our daily life as typical consumer devices, the performance of the proposed codec on the cellphone can indicate the practical performance on a variety of consumer devices. The CPU of the used cellphone is 1.5 GHz. The RAM is 2GB, while the ROM is 16GB. Moreover, all of the methods mentioned in experiments are complied with C++. The experiments consist of three parts. The first part is to verify the R-D performance, the second part compares the complexity of encoders, and the last part shows the energy consumption of each method. The detailed results are shown in the following sub-sections.

A. R-D Performances

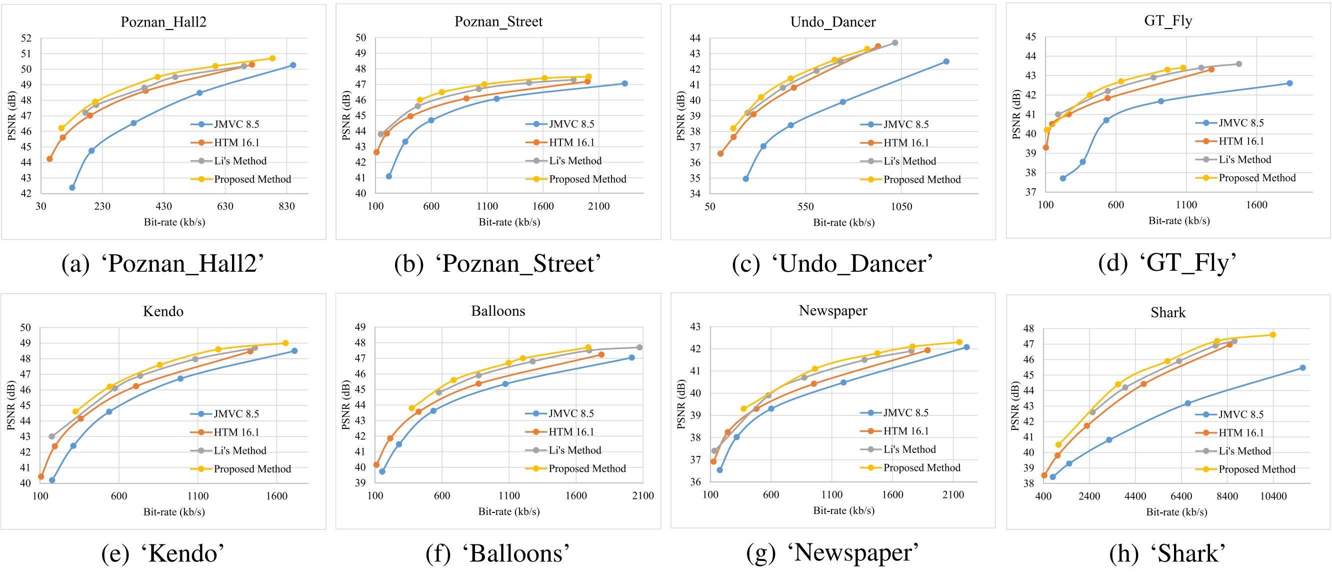
In the experiment, depth videos from two reference views are encoded, virtual view videos synthesized by original texture videos and coded depth videos are used to evaluate the R-D performance of the codec. The proposed codec is compared with 3D-HEVC (H.265) reference software HTM 16.1,



(a) ‘Poznan_Hall2’ (b) ‘Poznan_Street’ (c) ‘Undo_Dancer’ (d) ‘GT_Fly’

(e) ‘Kendo’ (f) ‘Balloons’ (g) ‘Newspaper’ (h) ‘Shark’

Fig. 6. R-D performances of ‘AllIntra’ mode. The proposed codec is compared with the method proposed by Li *et al.* [18], HTM 16.1 and JMVC 8.5.



(a) ‘Poznan_Hall2’ (b) ‘Poznan_Street’ (c) ‘Undo_Dancer’ (d) ‘GT_Fly’

(e) ‘Kendo’ (f) ‘Balloons’ (g) ‘Newspaper’ (h) ‘Shark’

Fig. 7. R-D performances of ‘IPPP’ mode. The proposed codec is compared with the method proposed by Li *et al.* [18], HTM 16.1 and JMVC 8.5.

and 3D-AVC (H.264) reference software JMVC 8.5. QPs of HTM and JMVC are set as 26, 31, 36, 41, 46. Meanwhile, the proposed method is also compared with the method propsed by Li *et al.* [18]. In [18], a GMM based compressive sensing video codec is proposed. In the codec, each block of the video is modeled by a GMM, and compressed by a product vector quantization. Moreover, the codec adopts differential pulse code modulation to reduce temporal redundancy.

For a comprehensive comparison, two coding conditions are adopted in the experiments. One is ‘AllIntra’ mode, and the other one is ‘IPPP’ mode. In ‘AllIntra’ mode, all frames are I frames, which means that only intra prediction is used to compress spatial redundancy and inter prediction schemes in all of the methods are disabled. In ‘IPPP’ mode, every 4 frames form a group, where the first frame is an I frame and following 3 successive frames are P frames in which temporal redundancy is also eliminated besides spatial redundancy.

The performance of ‘AllIntra’ mode are shown in Fig. 6. In the figure, x-axis implies the total bit-rate needed to encode left and right reference depth videos. y-axis implies the PSNR of the rendered virtual view synthesized by the coded depth videos and original texture videos, and PSNR is calculated by comparing the luminance component of the distorted frames and the original frames. It is obvious to see that the proposed method outperforms the other 3 methods in ‘AllIntra’ mode with a better R-D performance.

Fig. 7 shows the performance of 4 methods in ‘IPPP’ mode. The motion estimation and compensation schemes of 4 methods are enabled. In JMVC 8.5 and HTM 16.1, a full search based method is adopted to find the best reference

TABLE III
CODING EFFICIENCY COMPARISON OF 4 CODECS IN 'ALLINTRA' MODE

Sequences	**Measure**	**Depth Video Codecs**		
		Proposed Method	**Li *et al.* [18]**	**HTM 16.1**
Poznan_Hall2	BD-PSNR (dB)	**2.82**	2.31	1.40
	BD-Bitrate (%)	**-49.18**	-43.11	-33.77
Poznan_Street	BD-PSNR (dB)	**1.76**	1.47	0.61
	BD-Bitrate (%)	**-66.40**	-59.84	-31.41
Undo_Dancer	BD-PSNR (dB)	**3.59**	3.53	2.49
	BD-Bitrate (%)	**-61.50**	-60.15	-44.61
GT_Fly	BD-PSNR (dB)	**1.65**	1.43	0.99
	BD-Bitrate (%)	**-65.40**	-56.20	-39.88
Kendo	BD-PSNR (dB)	**1.65**	1.50	0.73
	BD-Bitrate (%)	**-38.66**	-33.50	-18.17
Balloons	BD-PSNR (dB)	**1.86**	1.46	0.76
	BD-Bitrate (%)	**-43.68**	-35.71	-19.97
Newspaper	BD-PSNR (dB)	**1.28**	1.00	0.41
	BD-Bitrate (%)	**-43.74**	-37.05	-17.59
Shark	BD-PSNR (dB)	**4.28**	3.91	2.16
	BD-Bitrate (%)	**-58.68**	-56.73	-43.75
Average	BD-PSNR (dB)	**2.36**	2.08	1.20
	BD-Bitrate (%)	**-53.41**	-47.78	-31.14

TABLE IV
CODING EFFICIENCY COMPARISON OF 4 CODECS IN 'IPPP' MODE

Sequences	**Measure**	**Depth Video Codecs**		
		Proposed Method	**Li *et al.* [18]**	**HTM 16.1**
Poznan_Hall2	BD-PSNR (dB)	**2.09**	1.83	1.61
	BD-Bitrate (%)	**-46.44**	-41.65	-37.54
Poznan_Street	BD-PSNR (dB)	**1.04**	0.95	0.65
	BD-Bitrate (%)	**-50.95**	-42.97	-34.07
Undo_Dancer	BD-PSNR (dB)	**2.99**	2.66	2.43
	BD-Bitrate (%)	**-56.34**	-50.57	-49.86
GT_Fly	BD-PSNR (dB)	**1.67**	1.35	1.25
	BD-Bitrate (%)	**-64.61**	-59.04	-53.60
Kendo	BD-PSNR (dB)	**1.25**	1.04	0.77
	BD-Bitrate (%)	**-32.74**	-27.75	-22.18
Balloons	BD-PSNR (dB)	**1.27**	0.97	0.59
	BD-Bitrate (%)	**-38.71**	-29.93	-20.21
Newspaper	BD-PSNR (dB)	**0.81**	0.73	0.39
	BD-Bitrate (%)	**-32.70**	-31.58	-19.14
Shark	BD-PSNR (dB)	**3.19**	2.87	2.17
	BD-Bitrate (%)	**-53.74**	-50.31	-48.41
Average	BD-PSNR (dB)	**1.79**	1.55	1.23
	BD-Bitrate (%)	**-47.03**	-41.73	-36.14

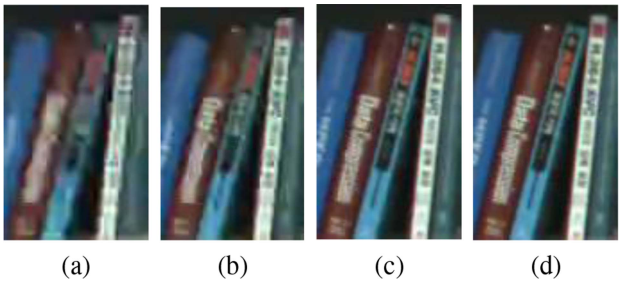


Fig. 8. Synthesized image of 3rd frame of sequence 'Newspaper' (QP=36) in 'AllIntra' mode. Note: (a) is encoded with JMVC 8.5, (b) is encoded with HTM 16.1, (c) is encoded with Li's method [18], (d) is encoded with the proposed method.

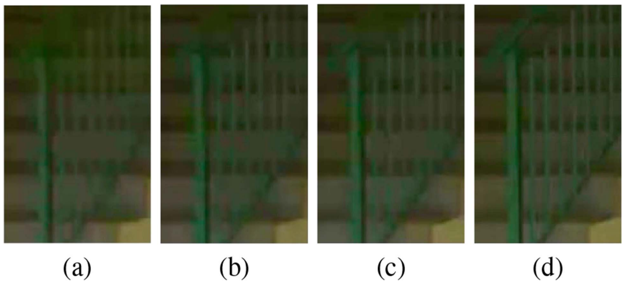


Fig. 9. Synthesized image of 14th frame of sequence 'Poznan_Hall2' (QP=31) in 'IPPP' mode. Note: (a) is encoded with JMVC 8.5, (b) is encoded with HTM 16.1, (c) is encoded with Li's method [18], (d) is encoded with the proposed method.

blocks. Li *et al.* [18] applied a differential pulse code modulation to remove temporal redundancy. Our method utilizes the proposed average value based fast motion estimation scheme.

In Fig. 7, x-axis is the total bit-rate needed to encode left and right reference depth videos, and y-axis is the PSNR of rendered virtual views. The R-D curves also show that our method outperforms the other 3 methods in 'IPPP' mode.

TABLE V
COMPLEXITY COMPARISON IN 'ALLINTRA' MODE

Sequences	**JMVC 8.5**	**Kim and Lee [11]**	**HTM 16.1**	**Wang *et al.* [10]**	**Li *et al.* [18]**	**Proposed method**
	Average time(sec)	**Average time(sec)**	**Average time(sec)**	**Average time(sec)**	**Average time(sec)**	**Average time(sec)**
Poznan_Hall2	710.56	545.78	11139.25	8430.44	465.52	**408.32**
Poznan_Street	764.95	610.52	13211.51	9894.26	444.92	**409.01**
Undo_Dancer	711.85	546.79	13220.83	10022.17	432.33	**404.88**
GT_Fly	753.90	600.41	13577.74	10157.87	458.07	**439.82**
Kendo	251.73	201.01	5507.79	4067.26	132.34	**119.97**
Balloons	262.10	203.91	5664.51	4224.66	134.46	**116.08**
Newspaper	288.16	266.32	5794.21	4352.88	152.80	**133.19**
Shark	770.87	635.02	14211.14	10660.46	472.08	**448.59**

TABLE VI
COMPLEXITY COMPARISON IN 'IPPP' MODE

Sequences	**JMVC 8.5**	**Kim and Lee [11]**	**HTM 16.1**	**Wang *et al.* [10]**	**Li *et al.* [18]**	**Proposed method**
	Average time(sec)	**Average time(sec)**	**Average time(sec)**	**Average time(sec)**	**Average time(sec)**	**Average time(sec)**
Poznan_Hall2	1763.45	1269.46	10161.74	7365.12	436.74	**412.30**
Poznan_Street	1913.02	1357.94	9726.24	7110.35	439.79	**380.19**
Undo_Dancer	1758.48	1230.43	9009.78	6484.72	429.42	**402.81**
GT_Fly	1652.75	1110.91	10565.29	7604.32	462.65	**446.29**
Kendo	757.97	528.95	6351.30	4573.65	136.24	**119.16**
Balloons	693.74	476.69	4345.09	3148.36	125.09	**115.96**
Newspaper	712.94	500.12	4167.00	2995.83	135.18	**124.42**
Shark	1976.61	1381.16	7605.84	5474.27	464.26	**440.85**

TABLE VII
ENERGY CONSUMPTION COMPARISON IN 'ALLINTRA' MODE

Sequences	**JMVC 8.5**	**Kim and Lee [11]**	**HTM 16.1**	**Wang *et al.* [10]**	**Li *et al.* [18]**	**Proposed method**
	Power cost(mAh)	**Power cost(mAh)**	**Power cost(mAh)**	**Power cost(mAh)**	**Power cost(mAh)**	**Power cost(mAh)**
Poznan_Hall2	162	149	3629	3295	92	**79**
Poznan_Street	174	160	4304	3798	88	**79**
Undo_Dancer	186	173	4307	3844	85	**78**
GT_Fly	197	181	4423	3987	92	**84**
Kendo	65	59	1794	1633	27	**23**
Balloons	68	63	1845	1663	27	**22**
Newspaper	75	69	1887	1693	31	**25**
Shark	171	156	4628	4094	96	**85**

TABLE VIII
ENERGY CONSUMPTION COMPARISON IN 'IPPP' MODE

Sequences	**JMVC 8.5**	**Kim and Lee [11]**	**HTM 16.1**	**Wang *et al.* [10]**	**Li *et al.* [18]**	**Proposed method**
	Power cost(mAh)	**Power cost(mAh)**	**Power cost(mAh)**	**Power cost(mAh)**	**Power cost(mAh)**	**Power cost(mAh)**
Poznan_Hall2	396	351	2749	2478	84	**76**
Poznan_Street	430	374	2637	2324	85	**71**
Undo_Dancer	395	348	2443	2223	83	**75**
GT_Fly	373	334	2869	2586	89	**82**
Kendo	171	150	1721	1535	27	**22**
Balloons	155	135	1179	1073	26	**21**
Newspaper	159	141	1136	1045	28	**22**
Shark	440	387	2073	1883	97	**88**

We also calculate the Bjontegaard-Delta (BD)-PSNR and the BD-bitrate [30] by taking the RD curves of JMVC 8.5 as the baseline. The detailed results for 'AllIntra' and 'IPPP' mode are shown in Table III and Table IV respectively. In the tables, a positive value for BD-PSNR indicates quality improvement of the compared codec than JMVC 8.5 , and a negative value for BD-bitrate indicates bitrate saving. The results shown in Table III and Table IV are consistent with the curves in Fig. 6 and Fig. 7. In 'AllIntra' mode, the proposed method, compared with JMVC 8.5, achieves 2.36 dB PSNR gain on average, and the bitrate saving achieves 53.41%. Compared with HTM 16.1 and Li *et al.* method [18], the proposed method has better performance both in BD-PSNR and BD-bitrate. For 'IPPP' mode, the proposed method also has better coding quality with BD-PSNR improvement up to 2.99 dB, compared with JMVC 8.5. The proposed method also has better performance than HTM 16.1 and Li *et al.* method [18].

Further, the subjective quality of the synthesized views is shown in Fig. 8 and Fig. 9. Fig. 8 is an example of 'AllIntra' mode, and Fig. 9 is an example of 'IPPP' mode. According to the figures, it is obvious to see that the proposed method, compared with other methods, has better subjective quality, especially at preserving boundaries.

In order to be adapted to the properties of depth videos, the proposed codec utilizes the adaptive wavelet decomposition to precisely separate high and low frequency. For high frequency, which indicates boundaries and details, a small QP is adopted to restrain the distortion of blocks. Depth videos, which maintain boundaries well, can naturally render the virtual view with high quality, and recover detail features of virtual view. Moreover, the proposed joint optimization scheme selects the best combination of QP and measurement rate for each block in order to further improve coding performance. Additionally, an average value based fast motion estimation is designed to reduce temporal redundancy, ensuring that the proposed encoder can achieve excellent performance in 'IPPP' mode.

B. Computational Complexity

In experiments, coding time is adopted to indicate coding complexity. The coding time is depicted as the average time consumed to encode depth videos on the cellphone, including 2 reference views and 5 bit-rate. Moreover, two methods are additionally added in the experiments to show the complexity of the proposed method, including the method proposed by Kim and Lee [11] and the method proposed by Wang *et al.* [10]. Both of two methods are low-complexity codecs applying in consumer devices. Kim and Lee [11] reduced complexity of AVC, while Wang *et al.* [10] proposed a low-complexity example of HEVC.

Table V shows the time comparison in 'AllIntra' mode. Compared with JMVC 8.5, HTM 16.1 and their low-complexity examples, the CS-based methods, including Li *et al.* method [18] and the proposed method, can save a lot of time due to the fact that it can generate sampling signals quickly. Compared with Li *et al.* method [18], the proposed method consumes less time, because the proposed adaptive DWT can decompose depth blocks efficiently with low complexity.

In 'IPPP' mode, Table VI demonstrates that the proposed method can save much more time than the traditional encoders, which means that the proposed method is with low-complexity. Compared with the motion estimation schemes in H.264 and H.265, the proposed method designs a fast motion estimation and compensation scheme. Compared with Li *et al.* method [18], P frames in the proposed method can obtain the decomposition tree T, QP, and measurement rate from the reference blocks, which saves a lot of time and reduces the complexity.

C. Energy Consumption

Considering that the proposed codec is applied in consumer devices which have limited energy, we also conduct experiments on the cellphone to show the energy consumption for the proposed method. Similar with the complexity, the average power consuming is used to evaluate the energy consumption. The average power consuming is depicted as the average power cost to encode depth videos on the cellphone, including 2 reference views and 5 bit-rate.

Table VII and Table VIII shows the power reduction of each method in 'AllIntra' mode and 'IPPP' mode respectively. It is obvious to see that the CS-based methods, including Li *et al.* method [18] and the proposed method, consume less power than JMVC 8.5, HTM 16.1 and their low-complexity examples. Compared with Li *et al.* method [18], the proposed method still saves more energy.

Different from traditional methods and their low-complexity examples, the CS-based methods cost less power due to the simple sampling process. Meanwhile, the proposed method designs a fast motion estimation algorithm which avoids complex full searching, saving a lot of energy. Thus, the proposed method reaches the least energy consumption in the compared methods. The low-complexity and energy-saving ensure that the proposed method can meet demands of consumer devices.

VII. Conclusion

In the paper, we proposed a low complexity CS-based depth video codec for consumer devices. The codec adopts an adaptive DWT algorithm to decompose depth blocks, and the algorithm follows the principle of local entropy minimization to reduce the local data volume. For temporal redundancy between successive frames, an average value based fast motion estimation scheme is designed to search the best matching block efficiently. Moreover, a joint optimization scheme is studied to determine the best combination of QP and measurement rate, in order to further improve the quality of encoded depth videos. The experimental results demonstrate that the proposed encoder, compared with H.264, can achieve BD-PSNR improvement up to 4.28 dB on virtual view videos in 'AllIntra' mode. In 'IPPP' mode, the proposed codec can achieve an average BD-PSNR gain of 1.79dB on virtual view videos. Compared with H.265 and the method proposed by Li *et al.* [18], the proposed codec also outperforms them in both 'AllIntra' mode and 'IPPP' mode. The subjective results show that the proposed codec can preserve the boundaries and detailed features of virtual view videos with high quality. Moreover, compared with H.264 and H.265, the complexity and the energy consumption of the proposed codec is much lower, which is adapted to consumer devices.

References

[1] Y. Song and Y.-S. Ho, "High-resolution depth map generator for 3D video applications using time-of-flight cameras," *IEEE Trans. Consum. Electron.*, vol. 63, no. 4, pp. 386–391, Nov. 2017.

[2] I. E. Richardson, *The H.264 Advanced Video Compression Standard*. Hoboken, NJ, USA: Wiley, 2010.

[3] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[4] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[5] A. Kammoun, W. Hamidouche, F. Belghith, J. F. Nezan, and N. Masmoudi, "Hardware design and implementation of adaptive multiple transforms for the versatile video coding standard," *IEEE Trans. Consum. Electron.*, vol. 64, no. 4, pp. 424–432, Nov. 2018.

[6] K. Singh and S. R. Ahamed, "Low power motion estimation algorithm and architecture of HEVC/H.265 for consumer applications," *IEEE Trans. Consum. Electron.*, vol. 64, no. 3, pp. 267–275, Aug. 2018.

[7] A. Singhadia, P. Bante, and I. Chakrabarti, "A novel algorithmic approach for efficient realization of 2D-DCT architecture for HEVC," *IEEE Trans. Consum. Electron.*, to be published.

[8] M. J. Garrido, F. Pescador, M. Chavarrías, P. J. Lobo, and C. Sanz, "A 2-D multiple transform processor for the versatile video coding standard," *IEEE Trans. Consum. Electron.*, to be published.

[9] B. Bross, J. Chen, and S. Liu, *Versatile Video Coding (Draft 5), V10*, document JVET-n1001 14th Meeting SG 16 WP3 and ISO/IEC JTC 1/SC 29/WG 11, ITU-T, Geneva, Switzerland, Mar. 2019.

[10] Y. Wang, X. Guo, X. Fan, Y. Lu, D. Zhao, and W. Gao, "Parallel in-loop filtering in HEVC encoder on GPU," *IEEE Trans. Consum. Electron.*, vol. 64, no. 3, pp. 276–284, Aug. 2018.

[11] H. Kim and H.-J. Lee, "A low-power surveillance video coding system with early background subtraction and adaptive frame memory compression," *IEEE Trans. Consum. Electron.*, vol. 63, no. 4, pp. 359–367, Nov. 2017.

[12] S. Li, L. D. Xu, and X. Wang, "Compressed sensing signal and data acquisition in wireless sensor networks and Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2177–2186, Nov. 2013.

[13] N. Cen, Z. Guan, and T. Melodia, "Multi-view wireless video streaming based on compressed sensing: Architecture and network optimization," in *Proc. 16th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2015, pp. 137–146.

[14] J. Duan, L. Zhang, Y. Liu, R. Pan, and Y. Sun, "An improved video coding scheme for depth map sequences based on compressed sensing," in *Proc. IEEE Int. Conf. Multimedia Technol.*, 2011, pp. 3401–3404.

[15] W.-S. Kim, S. K. Narang, and A. Ortega, "Graph based transforms for depth video coding," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2012, pp. 813–816.

[16] S. Lee and A. Ortega, "Adaptive compressed sensing for depthmap compression using graph-based transform," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2012, pp. 929–932.

[17] M. Sarkis and K. Diepold, "Depth map compression via compressed sensing," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2009, pp. 737–740.

[18] X. Li, X. Lan, M. Yang, J. Xue, and N. Zheng, "A new compressive sensing video coding framework based on Gaussian mixture model," *Signal Process. Image Commun.*, vol. 55, pp. 66–79, Jul. 2017.

[19] K. R. Vijayanagar, Y. Liu, and J. Kim, "Adaptive measurement rate allocation for block-based compressed sensing of depth maps," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2014, pp. 1307–1311.

[20] H.-W. Chen, L.-W. Kang, and C.-S. Lu, "Dynamic measurement rate allocation for distributed compressive video sensing," in *Proc. Visual Commun. Image Process. (VCIP)*, 2010, Art. no. 77440I.

[21] Y. Liu, M. Li, and D. A. Pados, "Motion-aware decoding of compressed-sensed video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 3, pp. 438–444, Mar. 2013.

[22] Y. Liu, K. R. Vijayanagar, and J. Kim, "Quad-tree partitioned compressed sensing for depth map coding," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2014, pp. 870–874.

[23] T. T. Do, Y. Chen, D. T. Nguyen, N. Nguyen, L. Gan, and T. D. Tran, "Distributed compressed video sensing," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2009, pp. 1393–1396.

[24] X. Zhang, A. Wang, B. Zeng, and L. Liu, "Adaptive distributed compressed video sensing," *J. Inf. Hiding Multimedia Signal Process.*, vol. 5, no. 1, pp. 98–106, 2014.

[25] N. Cen, Z. Guan, and T. Melodia, "Joint decoding of independently encoded compressive multi-view video streams," in *Proc. Picture Coding Symp. (PCS)*, 2013, pp. 341–344.

[26] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, vol. 12. Cham, Switzerland: Springer, 2013.

[27] T. Chan, S. Esedoglu, F. Park, and A. Yip, "Total variation image restoration: Overview and recent developments," in *Handbook of Mathematical Models in Computer Vision*. Boston, MA, USA: Springer, 2006, pp. 17–31.

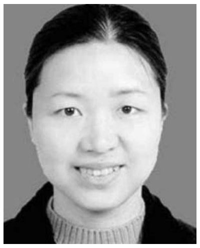
[28] A. Pinkus, *N-Widths in Approximation Theory*, vol. 7. Heidelberg, Germany: Springer, 2012.

[29] K. Müller and A. Vetro, *Common Test Conditions of 3DV Core Experiments*, document JCT3V-G1100, JCT-VC, San Jose, CA, USA, 2014.

[30] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document ITU-T Q. 6/SG16 15th Meeting, VCEG, Austin, TX, USA, Apr. 2001.

Shengwei Wang received the B.S. degree in telecommunication engineering from the Huazhong University of Science and Technology, Wuhan, in 2013, where he is currently pursuing the Ph.D. degree with the School of Electronic Information and Communications.

His research interests include 3-D video coding, machine learning, and related areas.



Li Yu (M'08) received the B.S., M.S., and Ph.D. degrees in electronic and information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1992, 1995, and 1999, respectively.

She is currently a Professor with the School of Electronic Information and Communications and the Director of the Research Center of Broadband Wireless Communication and Multimedia Communication, Huazhong University of Science and Technology. Her research interests include multimedia communication, wireless network, and signal processing in communications.

Sen Xiang (S'11–M'18) received the B.S. and Ph.D. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 2010 and 2016, respectively.

He was a Visiting Scholar with the University at Buffalo, State University of New York from September 2013 to August 2014. He joined the School of Information Science and Engineering, Wuhan University of Science and Technolgy in 2016, where he served as an Associate Professor. In 2017, he was selected as "Chutian Scholar". His research interests include depth image processing, 3-D video, and structured light depth acquisition and related areas. He has been a Reviewer of over 10 prestigious international journals from the IEEE, IET, and other associations.