INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 13, Number 2, Pages 205–219

ARBITRARY RESOLUTION VIDEO CODING USING COMPRESSIVE SENSING

HONG JIANG, CHENGBO LI, PAUL WILFORD, AND YIN ZHANG

Abstract. An arbitrary resolution video coding method based on compressive sampling is proposed. In this method, a video is coded using compressive measurements. The compressive measurements are made on videos of high resolution. The measurements may be used to reconstruct the video at the same resolution as the original video, and any subset of the measurements can be used to reconstruct video at lower resolution with a lower complexity. Video coding with arbitrary solution has important application in mobile video transmission.

Key words. Arbitrary resolution video coding, scalable video coding, compressive sampling, total variation, TV-DCT method, Kronecker product, Walsh-Hadamard transform.

1. Introduction

In a video network, a video source may be transmitted to multiple clients with different characteristics. The clients in the video network have different channel capacities, different display resolutions, and different computing resources. For example, a video source may be transmitted through the network to a high performance computer with a high resolution monitor in a residential home, and at the same time, to a mobile device with a low resolution screen and with a battery powered CPU. It is therefore desirable for a video source to be encoded in such a way that the same encoded video stream can be transmitted, and be usable by all clients, of different characteristics, in the network. In other words, we want to encode the video source once, but to transmit the same encoded video at different channel rates, and to decode it at different resolutions and with different complexities.

The traditional video coding such as MPEG2 does not provide the scalability desired for todays video network as described above. The lack of scalability exhibits itself in at least two ways. First, an encoded video is not scalable with transmission channel capacity. Because of its fixed bit rate, an encoded video stream is unusable in a channel supporting a lower bit rate, and at the same time, suboptimal in a channel with higher bit rate. This is the cause of the cliff effect encountered in video broadcast or multicast. Second, the MPEG2 video is not scalable with decoder resolution or decoding complexity. An encoded video can be decoded only at one resolution, with a fixed complexity (not considering post-processing such as resizing, or enhancement, after decoding). This creates the need for multiple encoded streams of the same video content to target decoders of different resolutions.

Efforts have been made to introduce scalability into video coding, noticeably by the scalable video coding (SVC) of H.264 [1] and the wavelet transform of Motion JPEG 2000 [2]. Both methods encode video into ordered layers, or levels, of streams, and the resolution, or quality, of the decoded video increases progressively as higher layers, or levels, are added to the decoder. Hierarchical modulation [3] may be used in conjunction with these scalable video codes to achieve more bandwidth

Received by the editors March 9, 2015.
2000 *Mathematics Subject Classification.* 68U10, 94A08.

efficiency. For example, the high priority of hierarchical modulation can be used to carry lower layers of the encoded video, and the low priority of hierarchical modulation can be used to carry the higher layers of the encoded video. These efforts have provided some alleviation to the problems such as the cliff effect in video transmission using the traditional video coding, but challenges of mobile video broadcast still remain. There has been an abundance of research activities in video coding to provide scalability in decoding resolution, see [4]-[7]. A joint video coding and transmission method was proposed in [8] to provide scalability with transmission channel capacity. These activities are in response to the fact that the scalability provided by H.264 or Motion JPEG 2000 is still not satisfactory. Specifically, the ordered layer structure does not provide scalability at a fundamental level, because a video encoded in these standards needs to be decoded at the lowest layer, and progressively built up to higher layers. The loss of a lower layer in the transmission makes the higher layers useless, even when they are received error-free. Therefore, the ordered layer structure is not scalable with the channel capacity [8].

Due to the proliferation of compressive sampling techniques [9],[19], video coding using compressive measurements is rapidly emerging [10]-[11]. Compressive video sensing offers the scalability desired in video network [12]-[13], and it is suitable for wireless transmission [14]. When the measurements of a video are made by a random (or pseudo-random) matrix, the video source information is distributed among the measurements of equal significance, and there are no measurements that are more important than others. The reconstruction of video requires a certain number of measurements to be available, but it does not need the availability of a particular subset of measurements. In this sense, a lost measurement due to transmission can simply be replaced by any other measurement. Further, since a video does not have a well defined sparsity, statistically, the more measurements are used in reconstruction, the better the quality of the reconstructed video gets [15]. If the measurements of the video are transmitted in broadcast or multicast, a receiver in a channel with higher capacity can have more measurements available, and hence a reconstructed video of higher quality, than a receiver in a channel with a lower capacity. These properties illustrate that video coding using compressive sampling is inherently scalable with the channel capacity, and it avoids the cliff effect in broadcast and multicast.

In this paper, we propose a framework for video coding using compressive measurements in which an encoded video is scalable both with the channel capacity and with decoding resolution and decoding complexity. Under the framework, a high resolution video is encoded using compressive measurements. The measurements are made once on the high resolution video. Any subset of the measurements can be used to reconstruct a video of same resolution as the oringal, or a lower resolution. The implication of this is very powerful in wireless transmission. The measurements from the high resolution video are transmitted in wireless broadcast/multicast network. A client in a good channel can correctly receive enough measurements to reconstruct a video of the original resolution with acceptable quality. A client in a poor channel may only correctly receive a subset with measurements fewer than required to reconstruct an acceptable video at high resolution, but the client may still use the correctly received measurements to reconstruct a video of a lower resolution, with an acceptable quality. The ability of arbitrary resolution reconstruction makes this video coding suitable for transmission in all channels.

Furthermore, a client in the network may be a handheld device with a small display and powered by a battery. It is undesirable for such a device to reconstruct

a video of the original size, and then resize it to a lower resolution for display, due to limit of power supply and computing resources. It is much more preferred that the device performs only the necessary processing to reconstruct a video of the lower resolution needed by the display with a reduced complexity.

Under this framework, a uniform encoding/multiscale decoding scheme is developed that provides low complexity reconstruction of video. The complexity and storage of decoding is proportional to the desired video resolution instead of the original resolution. In this scheme, the measurement matrices are constructed in a special way using the Kronecker products, which simplifies the reconstruction model, and reduces computation time. A multiscale compressive sampling scheme was also proposed in [16] to perform motion estimate and motion compensation during reconstruction. The paper is organized as follows. The framework for arbitrary resolution video coding is introduced in Section 2. The uniform encoding/multiscale decoding scheme is developed in Section 3. Numerical results will be presented in Section 4.

2. ARBITRARY RESOLUTION VIDEO CODING FRAMEWORK

In this section, the framework is developed in which a video source is first divided into video cubes. Then compressive measurements of the cubes are made. The reconstruction of a video cube is performed by solving a minimization problem. Various models and regularizations based on ℓ_1 and total variation (TV) [17] can fit into this framework. A TV-DCT method, minimizing the two dimensional total variation of the time domain discrete cosine transform, will be used due to its superior performance compared to other regularizations for video reconstruction [18]. The arbitrary resolution decoding is fulfilled by using an expansion matrix.

2.1. Video coding using compressive sampling. A source video consists of a number of frames of size $P \times Q$, where P and Q are the numbers of horizontal and vertical pixels in a frame, respectively. To encode it, the source video is divided into non-intersecting cubes. Each video cube consists of r frames of size $p \times q$. For the simplicity of discussions, every frame of a video cube is assumed to be taken from the same spatial region in its respective frame of the source video, although the framework still applies if each frame of video cube is taken from a different spatial region in its respective frame of source video, which could be done, for example, by using a motion estimate. Encoding is performed cube by cube on all video cubes making up the source video.

Let $x \in \Re^n$ be the vector obtained from a scan of the pixels of a video cube, i.e., x is a 1-D representation of the 3-D video cube, where $n = p \times q \times r$ is the length of the vector x. Normally, the pixels in a video cube, especially when the frames of the cube are chosen by a motion estimate scheme, are highly correlated, and therefore, vector x is sparse (having a small number of nonzero components) in some basis. This means that x can be well represented by using compressive measurements [9]. Let A be an $m \times n$ measurement matrix, then the m compressive measurements of x form the vector $y \in \Re^m$ defined by

$$y = Ax. \tag{1}$$

The measurements are considered to be the encoded values of the video cube. The encoding process is illustrated in Figure 1.

The measurement matrix A should be incoherent with the sparsity basis of the video cube, but in general, a random matrix can result in good performance [9]. In this paper, a permutated Walsh-Hadamard matrix will be used. This class of

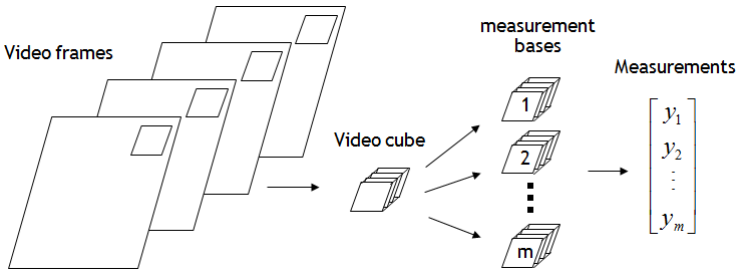

FIGURE 1. Video encoding using compressive measurements.

matrices can be easily implemented on hardware and they result in satisfactory recoverability.

2.2. Video reconstruction. Each video cube can be reconstructed from the measurements y by solving the following constraint minimization problem:

$$\min_x \Phi(x) \text{ subject to } y = Ax, \tag{2}$$

or in practice, the unconstraint problem

$$\min_x \Phi(x) + \frac{\mu}{2} \|Ax - y\|_2^2, \tag{3}$$

where $\Phi(x)$ represents the choice of regularization term and μ is the penalty parameter. The general theory on recoverability of a sparse signal from compressive measurements using a random matrix can be found, for example, in [9]. If the vector x is sparse, $\Phi(x)$ can be chosen to be the ℓ_1-norm of x [9]. However, when x is the vector made up of the pixels of a video cube, it is not obvious in which basis x is sparse, and further, in which basis, x has the most sparseness.

Total variation has been widely, and successfully, used as the regularization in image processing [15], [17]. As described in [18], we will use the spatial total variation of time domain DCT coefficients of the original cube as the regularization term, i.e.,

$$\Phi(x) = TV_s(DCT_t(x)). \tag{4}$$

In (4), $TV_s(z)$ is the 2D spatial total variation of the cube z defined as

$$TV_s(z) = \sum_{i,j,k} \sqrt{(z_{i+1jk} - z_{ijk})^2 + (z_{ij+1k} - z_{ijk})^2}, \tag{5}$$

for isotropic total variation, or

$$TV_s(z) = \sum_{i,j,k} |z_{i+1jk} - z_{ijk}| + |z_{ij+1k} - z_{ijk}|, \tag{6}$$

for anisotropic total variation.

Also in (4), $DCT_t(x)$ represents the pixelwise discrete cosine transform (DCT) of the video cube x in the temporal direction, and it is a cube in which each frame consists of DCT coefficients of a particular frequency. The minimization problem (3) is therefore to minimize the spatial total variation of the frequency components in time.

The minimization problem (3) is solved by the alternating minimization and augmented Lagrangian methods [18],[20]. The alternating minimization method

was first introduced to solve image deconvolution problem [21], which has a close relationship to (2) and (3).

2.3. Arbitrary resolution decoding. An advantage of using a random matrix, such as a permutated Walsh-Hadamard matrix, as the measurement matrix is that the measurements of the video are equally important, so that the quality of the reconstructed video only depends on the number of measurements available, independent of the availability of a particular measurement. It is this property that makes the coding inherently scalable. There is still more to be desired. Suppose the measurements of subsection 2.1 are transmitted, and, due to a low channel capacity, only very few measurements are correctly received at the decoder. The number of received measurements may be too small to reconstruct a video of the original resolution with an acceptable quality. It is possible to use the received measurements to reconstruct a video of the original resolution, and then resize it to a lower resolution, but the quality of the downsized video is inherently limited by that of the reconstructed video, although the smaller size of the downsized video may make some undesirable artifacts less obvious. Therefore, an alternative method is proposed in the following in which a video of lower resolution is reconstructed directly using the few measurements that are correctly received.

Assume a video cube of n pixels is k-sparse in certain basis, then the video cube can be reconstructed reliably with m measurements [9] if m satisfies

$$m \geq c \cdot k \cdot \log(n) \tag{7}$$

where c is some constant. It is reasonable to assume that the sparsity of a video is non-increasing as its resolution is lowered. Therefore, for a video cube of lower resolution, its number of pixels being n_L , and its sparsity being k_L , it is reasonable to assume

$$k_L \cdot \log(n_L) < k \cdot \log(n). \tag{8}$$

Equations (7) and (8) suggest that a lower resolution video has a better recoverability if the number of received measurements fails to satisfy condition (7) for the video of the original resolution. In other words, when there are too few measurements available to reconstruct a video with an acceptable quality, it is possible to use them to reconstruct a video of lower resolution with an acceptable quality. This analysis is confirmed by simulations to be given in a later section.

With the vector of measurements, y, for a source video cube given by (1), a lower resolution video cube can be reconstructed by using an expansion matrix. Formally, let E be an $n \times n_L$ matrix with full rank, where n_L is the number of pixels in the video cube of lower resolution, and $n_L < n$. Let $x_L \in \Re^{n_L}$ be the vector representing the video cube of the lower resolution. Then, x_L can be computed from the following minimization problem modified from (3):

$$\min_{x_L} \Phi(x_L) + \frac{\mu}{2} \left\| A \cdot E x_L - y \right\|_2^2 . \tag{9}$$

The expansion matrix E can be constructed by using any known resizing method. For example, matrix E can be constructed by using the DCT transform. Let T_n be the $n \times n$ matrix representing the DCT transform of size n , and $I_{n \times n_L}$ be the $n \times n_L$ matrix obtained from an $n_L \times n_L$ identify matrix by inserting $n - n_L$ rows of zeros. Then an expansion matrix is given by

$$E = T_n^T I_{n \times n_L} T_{n_L} . \tag{10}$$

Another example is to derive matrix E from a reduction matrix R , obtained from a video down-converting method. Let R be an $n_L \times n$ matrix representing the process of lowpass filtering and down-sampling, for example, by taking pixel averages, or using a poly-phase filter. It is also possible to construct the reduction matrix R from a 2D spatial DCT or wavelet transform of the frames of the video cube by using only the low frequency components. A well constructed reduction matrix R has the full rank, and therefore, the expansion matrix E can be obtained from the one-sided inverse of the reduction matrix R, as

$$E = R^T(RR^T)^{-1}. \tag{11}$$

Equations (1) and (9) constitute a video coding in which one encoding fits all channels and all display resolutions. This is illustrated in Figure 2(a).

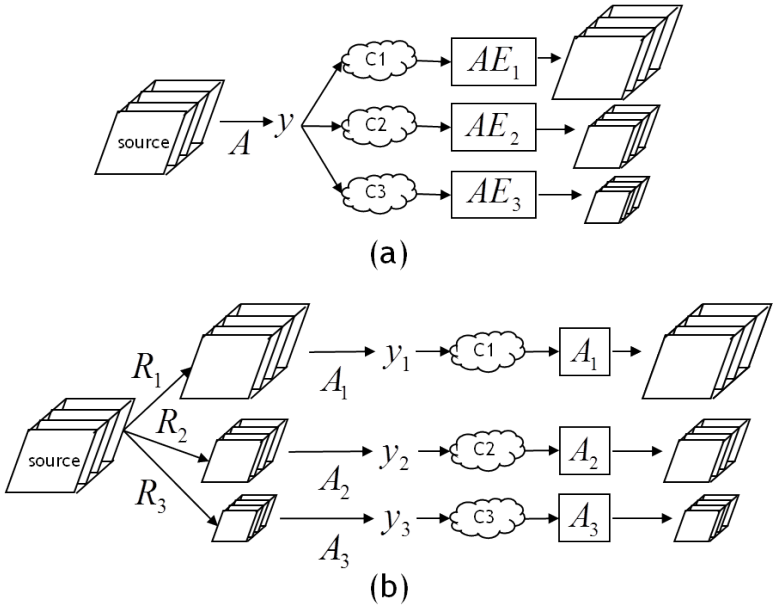

FIGURE 2. Video encoding, transmission and reconstruction.

In Figure 2(a), the source video is encoded using a random measurement matrix. The encoded video is transmitted in a broadcast system, and the correctly received measurements are used to reconstruct video of a desired resolution by using an appropriate expansion matrix E. More precisely, decoder i ($i = 1, 2, 3$) with channel capacity C_i may use an expansion matrix E_i to reconstruct a video of certain resolution by substituting $E = E_i$ in (9). An alternate, but unfavorable, encoding and transmission scheme is shown in Figure 2(b). In Figure 2(b), to transmit the source video to decoder i ($i = 1, 2, 3$) with channel capacity C_i, the source video is first down-sized to a resolution suitable for the display of the decoder by using a reduction matrix R_i. The down-sized video is encoded using a random matrix A_i. The compressive measurements are transmitted and the correctly received measurements are used to reconstruct the video of the same resolution as the down-sized video by substituting $A = A_i$ in (3). Clearly, the system in Figure 2(a) is more preferable. In the following, we will explore more the relationship of the systems in Figure 2(a) and Figure 2(b).

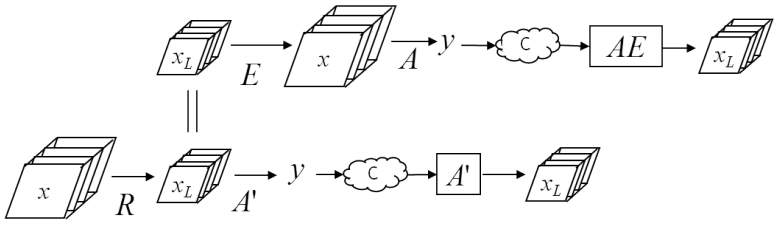

FIGURE 3. Coherence of expansion and reduction matrices.

Under the conditions as illustrated in Figure 3, the systems in Figure 2 are equivalent. Consider one branch (one channel and one decoder) in each of Figure 2(a) and Figure 2(b). If the down-sized video in Figure 2(b) can be expanded to the source video in Figure 2(a), as illustrated in Figure 3, then the reconstructed videos from Figure 2(a) and Figure 2(b) are identical provided that enough measurements are available at the decoders. More precisely, if the source video, the expansion matrix and the reduction matrix satisfy the following coherence condition,

$$x = ERx, \tag{12}$$

then equations (1) and (9) are equivalent to making measurements

$$y = A'(Rx), A' = A \cdot E \tag{13}$$

and reconstructing x_L by solving

$$\min_{x_L} \Phi(x_L) + \frac{\mu}{2} \|A'x_L - y\|_2^2 . \tag{14}$$

Equations (1) and (9) correspond to the top row of Figure 3, which in turn represent a branch of Figure 2(a), and equations (13) and (14) correspond to the bottom row of Figure 3, which in turn represent a branch of Figure 2(b).

Clearly, the solutions to (9) and (14) are the same, provided the theory of the compressive sampling applies, i.e., if there are enough measurements from the measurement matrix $A' = A \cdot E$ to recover the lower resolution video $x_L = Rx$.

Both (9) and (14) can be solved by the algorithm modified from TVAL3, which is an efficient TV minimization solver based on the alternating minimization and augmented Lagrangian methods for image reconstruction and denoising [20]. The detailed descriptions can be found in [18],[20].

3. UNIFORM ENCODING/MULTISCALE DECODING

The implementation of the framework described in section 2.3 may result in a very high complexity because of the evaluation of AEx_L. Unless the matrices are constructed with some special structures, either the complexity of AEx_L is proportional to that of the original resolution if the computation is performed as $A(Ex_L)$, or a large memory (to store the matrix AE) and a generic matrix-vector multiplication are required if the computation is performed as $(AE)x_L$. Therefore, it is highly desirable to simplify the computation of AEx_L for the mobile video application due to the limited resources available at decoders. In this section, an efficient scheme is proposed in which the sensing matrix is constructed with a special structure for encoding video uniformly and decoding at many lower resolutions.

We construct a measurement matrix A from the Kronecker product of small sensing matrices and structured permutation matrices. First, a predetermined number of decoding resolutions is specified. Each resolution will be called a level. Then, the

measurement matrix A is constructed for the specified number of decoding levels. The video is encoded by the compressive measurements of video cubes using the matrix A. The same measurements may be used to reconstruct a video of any one of the resolutions up to the lowest level specified. For this reason, the proposed method is named uniform encoding and multiscale decoding, because video of multiple resolutions can be reconstructed from the same encoded data. Specifically, A is constructed step by step as follows: 1. Specify the encoding level k , which determines the lowest resolution a video can be reconstructed from the encoded video. In other words, specify k so that the encoded video of the resolution $p \times q$ can be decoded to one of the resolutions $(p/2^l) \times (q/2^l)$, $l = 0, ..., k$. For the convenience of description, we always assume the dimensions $p/2^l$ and $q/2^l$ are integers. 2. Construct a series of permutation matrices $P_1^n, P_2^n, ..., P_k^n$, *named block-wise vectorized permutations*

$$P_i^n = P_{i-1}^{n/4} \otimes I_4 \text{ for } 1 < i \leq k, \tag{15}$$

where $P_i^s \in \Re^{s \times s}$ and I_4 represents the 4×4 identity matrix. Initially, P_1^s is the vectorized permutation based on 2×2 blocks. For example, P_1^{16} is the permutation matrix that works in the way illustrated in Figure 4.

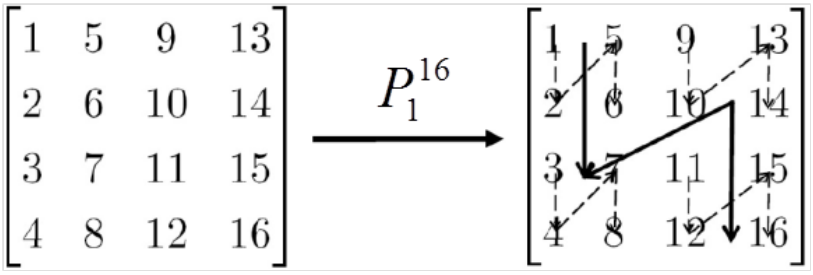

FIGURE 4. Definition of permutation matrix.

In other words, let $u = [1, 2, 3, 4, 5, 6, 7, 8, ..., 13, 14, 15, 16]^T$ be the column vector formed by concatenating the columns of the matrix on the left hand side of Figure 4. Then $P_1^{16}u = [1, 2, 5, 6, 3, 4, 7, 8, ..., 11, 12, 15, 16]^T$. In general, for a matrix U of a dimension with n entries, u is the column vector formed by concatenating the columns of the matrix U, and $P_1^n u$ is the column vector formed by first dividing the matrix U into blocks of four elements (2×2 blocks), and then concatenating the columns of each 2×2 block followed by concatenating all these 2×2 blocks column by column.

From this point on, we will omit the superscript of P_i^s for simplicity. Its size can be determined by properly forming matrix products. 3. Select a series of small sensing matrices $A_0 \in \Re^{m_0 \times (n/4^k)}$ and $A_i \in \Re^{m_i \times 4}$ for $1 \leq i \leq k$, which satisfy

$$\prod_{i=0}^{k} m_i = m \text{ and } 4 \geq m_1 \geq \ldots \geq m_k \geq 1. \tag{16}$$

The choice of $m_0, m_1, \ldots, m_k$ is not unique, but we should choose m_0 equal to $n/4^k$ or as large as possible to guarantee the recoverability at the relatively low resolution.

4. Let

$$Q_k = P_k P_{k-1} \cdots P_1 \tag{17}$$

$$A = (A_0 \otimes A_1 \otimes \cdots \otimes A_k)Q_k, \tag{18}$$

which gives the measurement matrix for uniform encoding/multiscale decoding scheme.

This structured measurement matrix can lead to a significant reduction of decoding complexity. Some notations will be introduced before getting into details.

Level $l(l \leq k)$ decoding refers to the resolution of the reconstructed video cube being $(p/2^l) \times (q/2^l)$ and $U_l \in \Re^{(p/2^l)\times(q/2^l)\times r}$ denotes the level l resolution approximation of a video cube U. In other words, U_l is the video having a resolution of $(p/2^l) \times (q/2^l)$ reconstructed from the compressive measurements of the original video cube U of the resolution $p \times q$. Vectors x and x_l represent the vectorizations of U and U_l, respectively, by concatenating the pixels of video cubes column by column, and then, frame by frame. Furthermore, $1_{s\times t}$ represents an $s \times t$ matrix whose entries are 1 everywhere. The second dimension of subscript t can be omitted if $t = 1$. $B^{\circ j}$ denotes the j-degree power of Kronecker product, i.e.,

$$B^{\circ j} = \underbrace{B \otimes \cdots \otimes B}_{j}. \tag{19}$$

One way to approximate U is using

$$U \approx U_l \otimes 1_{2^l \times 2^l}, \tag{20}$$

which is equivalent to

$$P_l \cdots P_1 x \approx x_l \otimes 1_{4^l} = x_l \otimes 1_4^{\circ l}. \tag{21}$$

Therefore, we can define the expansion matrix E as follows:

$$Ex_l = P_1^T \cdots P_l^T (x_l \otimes 1_4^{\circ l}). \tag{22}$$

Then, we have

$$AEx_l = A(P_1^T \cdots P_l^T (x_l \otimes 1_4^{\circ l})). \tag{23}$$

Combining (23) with (17) and (18), we can derive

$$\begin{aligned}
AEx_l &= (A_0 \otimes A_1 \otimes \cdots \otimes A_k) \\
&\cdot Q_k(P_1^T \cdots P_l^T (x_l \otimes 1_4^{\circ l})) \\
&= (A_0 \otimes A_1 \otimes \cdots \otimes A_k) \\
&\cdot P_k \cdots P_{l+1}(x_l \otimes 1_4^{\circ l}) \\
&= (A_0 \otimes A_1 \otimes \cdots \otimes A_k)((P_{k-l} \otimes I_4^{\circ l}) \\
&\cdots (P_1 \otimes I_4^{\circ l})(x_l \otimes 1_4^{\circ l})) \\
&= (A_0 \otimes A_1 \otimes \cdots \otimes A_k) \cdot ((P_{k-l} \cdots P_1 x_l) \\
&\otimes (I_4^{\circ l} \cdots I_4^{\circ l} 1_4^{\circ l})) \\
&= ((A_0 \otimes \cdots \otimes A_{k-l}) \otimes A_{k-l+1} \cdots \otimes A_k) \\
&\cdot ((P_{k-l} \cdots P_1 x_l) \otimes 1_4^{\circ l}) \\
&= ((A_0 \otimes \cdots \otimes A_{k-l})P_{k-l} \cdots P_1 x_l) \\
&\otimes (A_{k-l+1} 1_4) \cdots \otimes (A_k 1_4).
\end{aligned} \tag{24}$$

Let $L_k^j = (A_0 \otimes \cdots \otimes A_{k-l})P_{k-l} \cdots P_1$ and $a_j = A_j 1_4$ for $j \leq k$. Then the minimization problem (9) is equivalent to the following level l decoding model:

$$\min_x \Phi(x_l) + \frac{\mu}{2} \left\| (L_k^l x_l) \otimes a_{k-l+1} \otimes \cdots \otimes a_k - y \right\|_2^2. \tag{25}$$

The low resolution video cube x_l can be obtained by solving the minimization problem (25).

TVAL3 has been proven as an efficient solver for 2D TV minimization problem and can be extended to handle higher dimensional problems [18]. We choose to extend TVAL3 algorithm to solve (25) for decoding. The complexity of this algorithm is dominated by two matrix-vector multiplications at each iteration, which is proportional to the size of L_k^l. As a matter of fact, $L_k^l \in \Re^{(\prod_{i=0}^{k-l} m_i)\times(n/4^l)}$ corresponds to the desired resolution $(p/2^l) \times (q/2^l)$ instead of the original resolution $p \times q$. Therfore, the uniform encoding/multiscale decoding scheme is able to provide low complexity and decoding time is scalable with the resolution of the reconstructed video.

4. SIMULATION

The coding method described in section 3 is implemented in simulations using an encoding matrix that is capable of providing three levels of decoded resolution. Those small sensing matrices for the construction of A are extracted from the permutated Walsh-Hadamard matrices. Results for three standard video test sequences will be presented, and they are Container, Hall and News. All three source video sequences are of CIF resolution (352×288 pixels/frame) at 30 frames per second (fsp).

For each source video, the same measurement matrix as described in section 3 is used to encode the video. Each video cube consists of 8 entire frames of size 352×288. That is, the number of pixels in a source video cube is

$$n = 352 \times 288 \times 8 = 811008.$$

For each source video, decoding of three resolutions are performed: the original CIF resolution (352×288), the QCIF resolution (176×144) and the QQCIF resolution (88×72). A different amount of measurements are used in the reconstructions of video with a different resolution. Let m be the number of measurements used in the reconstruction. For all three source video sequences, $m = 0.35 \cdot n$ (35% measurements) is used for the CIF reconstructions, $m = 0.09 \cdot n$ (9% measurements) is used for the QCIF reconstructions and $m = 0.01 \cdot n$ (1% measurements) is used for the QQCIF reconstructions. Figures 4-6 show the typical results.

The complexity of the reconstruction is scalable with the resolution of the decoded pictures. This is evident from the CPU time it takes to decode the video of different resolutions. When the average time it takes to decode a video of CIF resolution is normalized to 1, the average time it takes to decode a video of QCIF resolution is .22, and the average time it takes to decoder a video of QQCIF is .046. Next, the accuracy in the reconstructions will be measured by using PSNR in the reconstructed video. In order to measure the PSNR, a reconstructed video must be compared with an original video of the same resolution. To accomplish this, a reference video is first resized to a higher resolution to be used as the source video. Then the source video is encoded and decoded. The decoded video has the same resolution as the reference video. Finally, the PSNR of the decoded video as compared to the reference video is measured and reported. Three methods are used in the simulations and the PSNR of the decoded video from the three methods will be reported. These methods are illustrated in Figure 8.

Figure 8. Three methods used for PSNR calculation: (a) uniform encoding/multiscale decoding (UEMD) of this paper, (b) conventional compressive sampling reconstruction followed by resizing to lower resolution and (c) the 3D DCT method followed by resizing to lower resolution.

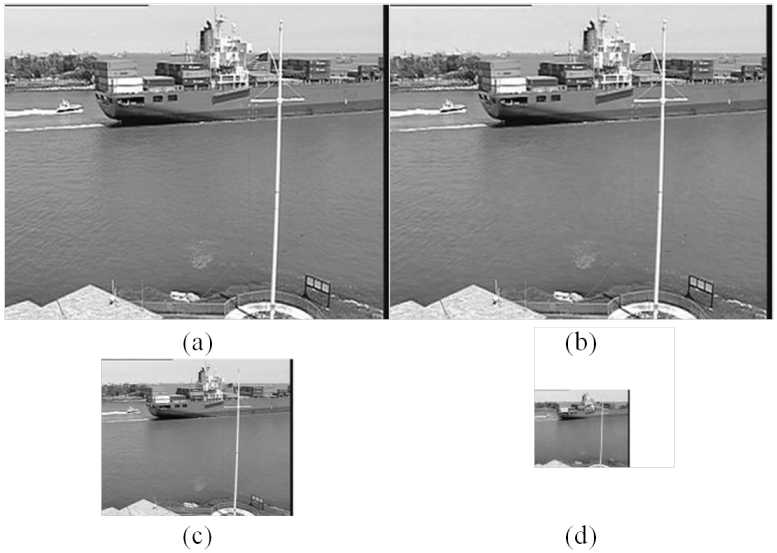

FIGURE 5. Frame 4 of Container video clip: (a) original CIF frame, (b) the reconstructed CIF frame using 35% of measurements, (c) the reconstructed QCIF frame with 9% of measurements and (d) the reconstructed QQCIF frame with 1% of measurements.

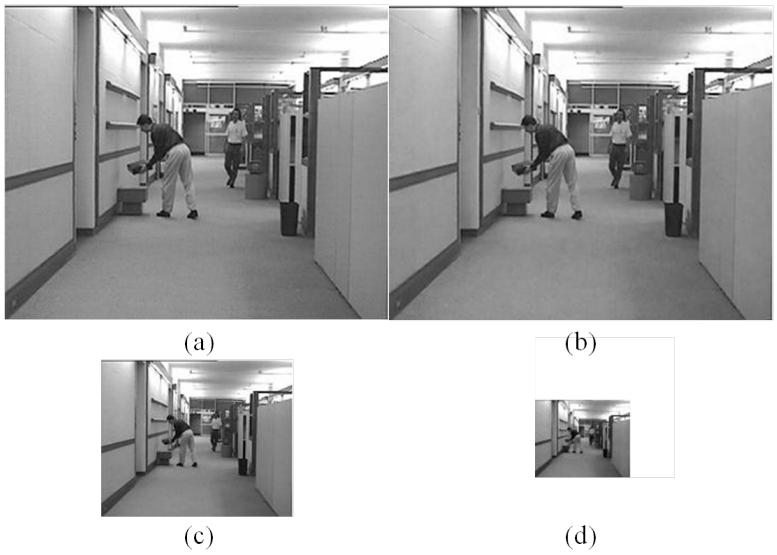

FIGURE 6. Frame 4 of Hall video clip: (a) original CIF frame, (b) the reconstructed CIF frame using 35% of measurements, (c) the reconstructed QCIF frame with 9% of measurements and (d) the reconstructed QQCIF frame with 1% of measurements.

In Figure 8, a reference video x_R is converted to the source video x by an expansion matrix E , i.e., $x = Ex_R$. The same expansion is used for all methods.

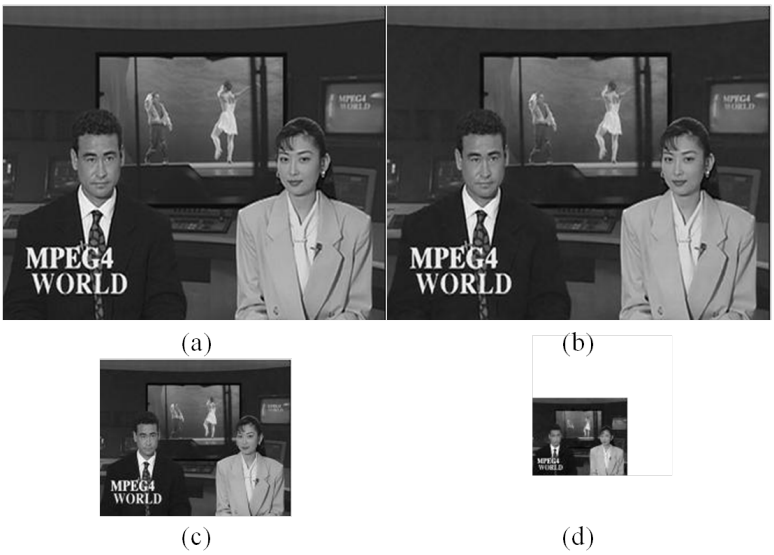

FIGURE 7. Frame 4 of News video clip: (a) original CIF frame, (b) the reconstructed CIF frame using 35% of measurements, (c) the reconstructed QCIF frame with 9% of measurements and (d) the reconstructed QQCIF frame with 1% of measurements.

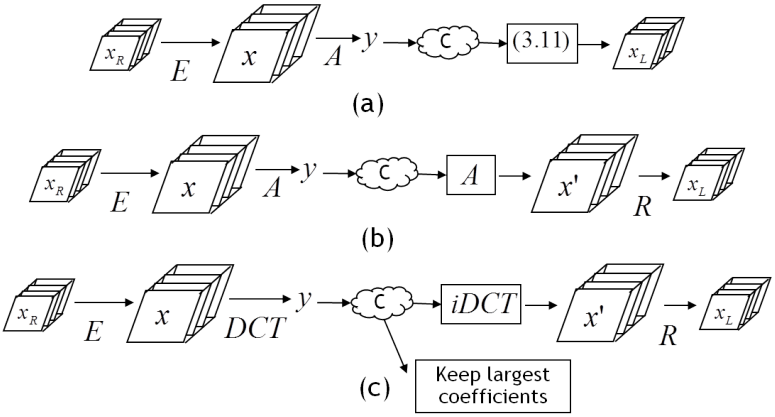

FIGURE 8. Three methods used for PSNR calculation: (a) uniform encoding/multiscale decoding (UEMD) of this paper, (b) conventional compressive sampling reconstruction followed by resizing to lower resolution and (c) the 3D DCT method followed by resizing to lower resolution.

Specifically, the source video x is obtained from the reference video x_R by duplicating the pixels of x_R . The source video is then encoded, transmitted and decoded by three different methods. The first method is the method of this paper (UEMD) as shown in Figure 8(a). The lower resolution decoded video x_L is obtained directly as part of reconstruction from correctly received measurements y by solving (3.11). The second, shown in Figure 8(b), is a conventional compressive sampling reconstruction. The measurement matrix A is a permutated Walsh-Hadmard matrix.

The correctly received measurements y are used to reconstruct a video x' of the same resolution as the source video by solving (2.3). Then the reconstructed video is resized to the lower resolution x_L by taking the average of the pixels of x' . The last, shown in Figure 8(c), is the 3D DCT method. The source video x is encoded by 3D DCT transform on a video cube. The DCT coefficients are transmitted. The correctly received y are the largest coefficients of DCT transform. In other words, the coefficients are sorted in descending order according to their amplitudes. For example, if 10% coefficients are received, it is assumed that the first 10% of the sorted coefficients (the largest 10% in amplitudes) are received correctly. This, of course, places a huge advantage to the DCT method, because in the compressive sampling methods of Figure 8(a) and Figure 8(b), the correctly received measurements are randomly chosen. In all methods, the PSNR is calculated by comparing x_L with the reference video x_R. The PSNR values as a function of the percentage of measurements received for the video clip Hall are shown in Figure 9 and Figure 10.

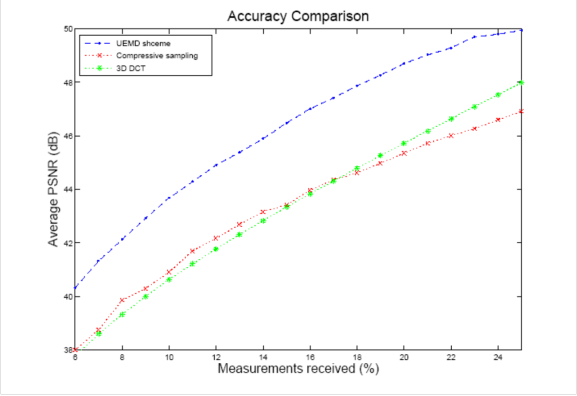

FIGURE 9. PSNR for video clip Hall: source video x is CIF (352x288) and the decoded video x_L is QCIF (176x144).

In Figure 9, the source video clip Hall of CIF resolution is encoded and transmitted as previously described. The decoded video has QCIF resolution. The decoded video has half as many pixels as the source video in both horizontal and vertical directions. The QCIF reference video is obtained by taking averages of two adjacent pixels in both horizontal and vertical directions. The dashed blue curve is the PSNR for the method of this paper (UEMD) and the red curve with crosses is the PSNR for the conventional compressive sampling reconstruction, and the green curve with squares is the PSNR for the 3D DCT method.

In Figure 10, the reference video is also Hall of QCIF resolution. The source video of 4CIF resolution is obtained from the CIF video by repeating the pixels of the CIF video. The decoded video has QCIF resolution. The decoded video has 1/4 as many pixels as the source video in both horizontal and vertical directions. The dashed blue curve is the PSNR for the method of this paper (UEMD) and the red curve with crosses is the PSNR for the conventional compressive sampling reconstruction, and the green curve with squares is the PSNR for the 3D DCT method. The results in Figure 9 and Figure 10 show that the method proposed in this paper has better accuracy than the methods in which a video of the original resolution is reconstructed and then resized to a lower resolution.

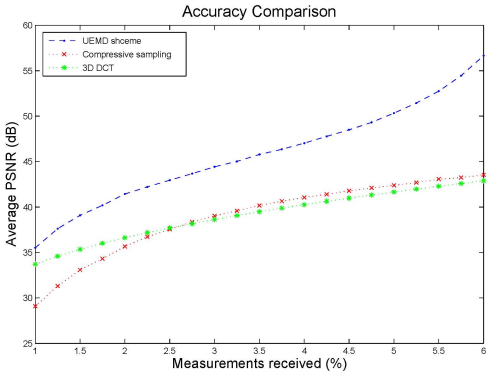

FIGURE 10. PSNR for video clip Hall: source video x is 4CIF (704x576) and the decoded video x_L is QCIF (176x144).

5. Conclusion

The video coding framework of this paper provides full scalability for both channel capacity and display resolutions. The complexity and running time of new method is also scalable based on different desired resolutions. Simulation results demonstrate that the uniform encoding/multiscale decoding scheme has a better performance than the traditional reconstruction followed by resizing. The property that one encoding fits all resolutions has importation application in mobile video communications.

References

[1] H. Schwarz, D. Marpe, and T. Wiegand, Overview of the scalable video coding extension of H.264/AVC, IEEE Trans. Circuits and Systems for Video Tech., vol. 17, no. 9, pp. 1103-1120, Sept. 2007.

[2] D.S. Taubman and M.W. Marcellin, JPEG 2000 Image Compression Fundamentals, Standards and Practice, Kluwer Academic Publishers, The Netherlands 2001.

[3] H. Jiang and P. Wilford, A Hierarchical Modulation for Upgrading Digital Broadcast Systems, IEEE Ttrans. Broadcasting, vol. 51, no. 2, pp.223-229, June 2005.

[4] J. Vass and X. Zhuang, Multiresolution-multicast video distribution over the Internet, 2000 IEEE Wireless Communications and Networking Conference, pp. 1457 - 1461, Sep 2000.

[5] C. Li 1, H. Xiong, J. Zou, T. Chen, A Unified QoS Optimization for Scalable Video Multirate Multicast over Hybrid Coded Network, 2010 IEEE International Conference on Communications (ICC), pp. 23-27, May 2010.

[6] C. Mairal and M. Agueh, Smooth and Scalable Wireless JPEG 2000 Images and Video Streaming with Dynamic Bandwidth Estimation, 2010 Second International Conferences on Advances in Multimedia (MMEDIA), pp 174 - 179, 2010.

[7] J. Xu, X. Shen, J.W. Mark, and J. Cai, Adaptive transmission of multilayered video over wireless fading channels, IEEE Trans. on Wireless Communications, vol 6, no 6 pp. 2305-2314, June 2007.

[8] S. Jakubczak, H. Rahul and D. Katabi, SoftCast: One video to serve all wireless receivers, Computer Science and Artificial Intelligence Laboratory Technical Report, MIT-CSAIL-TR-2009-005, MIT, Feb, 2009.

[9] E. Candes, J. Romberg, and T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, IEEE Trans. on Information Theory, vol 52, no 2, pp. 489-509, Feb 2006.

[10] M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, Compressive imaging for video representation and coding, Proc. Picture Coding Symposium (PCS), Beijing, China, April 2006.

[11] V. Stankovic, L. Stankovic, and S. Cheng, Compressive video sampling, European Signal Processing Conf. (EUSIPCO), Lausanne, Switzerland, August 2008.
[12] T. Do, Y. Chen, D. Nguyen, N. Nguyen, L. Gan and T. Tran, Distributed compressed video sensing, 16th IEEE International Conference on Image Processing (ICIP), pp. 1393 - 1396, 2009.
[13] J. Prades-Nebot, Y. Ma and T. Huang, Distributed Video Coding using Compressive Sampling 2009 Picture Coding Symposium, PCS 2009, pp 1-4, 2009.
[14] S. Pudlewski and T. Melodia, On the Performance of Compressive Video Streaming for Wireless Multimedia Sensor Networks, 2010 IEEE International Conference on Communications (ICC), 2010.
[15] J. Romberg, Imaging via compressive sampling, IEEE Signal Processing Magazine, vol 25, no 2, pp. 14 - 20, March 2008.
[16] J.Y. Park and M.B. Wakin, A Multiscale Framework for Compressive Sensing of Video, Picture Coding Symposium (PCS), Chicago, Illinois, May 2009.
[17] L. Rudin, S. Osher, and E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D, pp. 259 - 268, 1992.
[18] C. Li, H. Jiang, and P. Wilford and Y. Zhang, A new compressive video sensing framework for mobile communications, in preparation.
[19] D Donoho, Compressed sensing, IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289 - 1306, 2006.
[20] C. Li, An Efficient Algorithm for Total Variation Regularization with Applications to the Single Pixel Camera and Compressive Sensing, Mater Thesis, Computational and Applied Mathematics, Rice University, 2009.
[21] Y. Wang, J. Yang, W. Yin, and Y. Zhang, A new alternating minimization algorithm for total variation image reconstruction, SIAM J. Imag. Sci., vol. 1, no. 4, pp. 248 - 272, 2008.

Bell Labs Alcatel-Lucent 700 Mountain Ave Murray Hill, NJ 07974
E-mail: hong.jiang@alcatel-lucent.com