Distributed video coding based on sampling of signals with finite rate of innovation

Varit Chaisinthop and Pier Luigi Dragotti

Electrical and Electronic Engineering
Imperial College London,
Exhibition Road, London SW7 2AZ, United Kingdom

ABSTRACT

This paper proposes a new approach to distributed video coding. Distributed video coding is a new paradigm in video coding, which is based on the concept of decoding with side information at the decoder. Such a coding scheme employs a low-complexity encoder, making it well suited for low-power devices such as mobile video cameras.

The uniqueness of our work lies in the combined use of discrete wavelet transform (DWT) and the concept of sampling of signals with finite rate of innovation (FRI).[1] This enables the decoder to retrieve the motion parameters and reconstruct the video sequence from the low-resolution version of each transmitted frame. Unlike the currently existing practical coders, we do not employ traditional channel coding technique. For a simple video sequence with a fixed background, our preliminary results show that the proposed coding scheme can achieve a better PSNR than JPEG2000-intraframe coding at low bit rates.

Keywords - distributed video compression, wavelet, FRI signals, sampling, moments.

1. INTRODUCTION

The rapid growth in the area of "uplink" rich media applications in today's emerging era of mobile devices has sparked the interest in the development of practical distributed coding algorithms[2].[3] The theoretical foundation, however, dates back to the information theoretic results of Slepian and Wolf[4] and Wyner and Ziv[5] of the 1970s. Slepian and Wolf[4] gave the achievable rate region for the lossless encoding of correlated sources while some extensions to the lossy case were presented in.[5] These results can be used to develop video coding schemes, which use a low-complexity encoder while achieve a better compression efficiency than the "intraframe" coding. This is made possible by shifting the load of computational complexity to the decoder side and the "interframe" dependency of the video sequence is exploited at the receiver. This unconventional balance in complexity is, in fact, a part of the architectural requirements of the "uplink" rich media applications.[3]

Existing distributed video coders use sophisticated channel codes to reconstruct the video sequence at the decoder (see[2] for a comprehensive review). In this paper, we investigate the use of discrete wavelet transform (DWT) together with the concept of sampling of signals with finite rate of innovation (FRI)[1] to implement motion estimation at the decoder. We present three coding schemes for the following scenarios: (a) a polygon moving by translation in a uniform background; (b) the extension of (a) to the case where motion can be described by an affine transform; (c) a real video sequence with a fixed background. The problem of bit allocation as well as quantization will also be discussed.

In the next section, main results of sampling of FRI signals are discussed. Our proposed distributed coding schemes for the three scenarios are presented in Section 3. In Section 4, we discuss the issue of quantization in our proposed schemes. The preliminary results are given in Section 5. Finally, conclusions are drawn in Section 6.

Further author information: (Send correspondence to V.Chaisinthop)
V.Chaisinthop: E-mail: varit.chaisinthop02@imperial.ac.uk
P.L.Dragotti: E-mail: p.dragotti@imperial.ac.uk

2. SAMPLING OF 2-D FRI SIGNALS

Recent developments in sampling theory have focussed on classes of non-band limited signal, one of which is a class of signals with finite rate of innovation (FRI).[6] The definition and sampling schemes of FRI signals are given in details in.[1] In this section, we will only discuss the main results that will be used in the sequel. The family of sampling kernel used in our setup includes functions that reproduce polynomials and thus satisfy the Strang-Fix conditions.[1]

Let a 2-D continuous signal be $f(x, y)$ with $x, y \in \mathbb{R}$ and let the 2-D sampling kernel be $\varphi(x, y)$. In a typical sampling setup, the samples obtained by sampling $f(x, y)$ with $\varphi(x, y)$ are given by:

$$S_{m,n} = \langle f(x,y), \varphi(x/T - m, y/T - n) \rangle \,, \tag{1}$$

where $\langle \cdot \rangle$ denotes the inner product and $m, n \in \mathbb{Z}$. Assume that the sampling kernel satisfies the polynomial reproduction property i.e.:

$$\sum_{m \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} c_{m,n}^{(p,q)} \varphi(x - m, y - n) = x^p y^q \tag{2}$$

with $p, q \in N$ and a proper set of coefficients $c_{m,n}^{(p,q)}$. It follows that, with $T = 1$, the continuous geometric moment $m_{p,q}$ of order $(p + q)$ of the signal $f(x, y)$ is given by:

$$\begin{aligned}
m_{p,q} &= \int\int f(x,y) x^p y^q dxdy \\
&= \left\langle f(x,y), \sum_{m \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} c_{m,n}^{(p,q)} \varphi(x-m, y-n) \right\rangle \\
&= \sum_{m \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} c_{m,n}^{(p,q)} \langle f(x,y), \varphi(x-m, y-n) \rangle \\
&= \sum_{m \in \mathbb{Z}} \sum_{n \in \mathbb{Z}} c_{m,n}^{(p,q)} S_{m,n}.
\end{aligned} \tag{3}$$

Therefore, given a set of coefficients $c_{m,n}^{(p,q)}$, one can retrieve the continuous moments from an arbitrarily low-resolution set of samples $\widehat{S}_{m,n}$ provided that $f(x, y)$ lies in the region where equation (2) is satisfied.

The fact that any valid scaling function will reproduce polynomials is well known. We use the scaling function $\varphi_j(x, y)$ of the DWT as a sampling kernel in our coding scheme, where j represents the number of iteration of the wavelet decomposition. Here, $\varphi_j(x, y)$ is given by the tensor product of two 1-D scaling functions $\varphi_{j,n}(t) = 2^{-j/2}\varphi(2^{-j}t - n)$, $j, n \in \mathbb{Z}$. Thus we can change the resolution of $S_{m,n}$ by altering j with the corresponding sampling period of $T = 2^j$. It then follows that the required coefficients $c_{m,n}^{(p,q)}$ are given by:

$$\begin{aligned}
\left\langle x^p y^q, \widetilde{\varphi}_j(x-m, y-n) \right\rangle &= \sum_{k \in \mathbb{Z}} \sum_{l \in \mathbb{Z}} c_{k,l}^{(p,q)} \left\langle \varphi_j(x-k, y-l), \widetilde{\varphi}_j(x-m, y-n) \right\rangle \\
&= \sum_{k \in \mathbb{Z}} \sum_{l \in \mathbb{Z}} c_{k,l}^{(p,q)} \delta_{k-m,l-n} \\
&= c_{m,n}^{(p,q)}
\end{aligned} \tag{4}$$

where $\widetilde{\varphi}_j(x, y)$ is the dual of $\varphi_j(x, y)$. In the next section, we show that the motion parameters can be extracted from the set of continuous moments $m_{p,q}$ obtained using equation (3).

3. A NOVEL APPROACH TO DISTRIBUTED VIDEO CODING

We now present our proposed distributed coding schemes, which are based on the concept of sampling of FRI signals, for the three scenarios stated above. The basic framework behind our coding scheme is as follows; the first frame of the video sequence is treated as the "key frame" and is encoded with a conventional intraframe

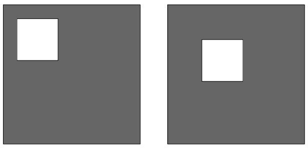

Figure 1. A bilevel polygon moving by translation in a uniform background

coding method. The rest of the frames are "non-key frames" and are sampled by a kernel that reproduces polynomial. The decoder decodes the key frame and retrieves the continuous moments from the samples. The motion parameters are calculated using these moments. Lastly, the non-key frames are reconstructed using the key frame and the obtained motion parameters. Compression is thus achieved by reducing the resolution of the samples. Since the encoder does not perform any interframe coding or motion estimation, its complexity is much lower than that of the decoder.

3.1. A bi-level polygon moving by translation

In order to gain some intuition, we start by considering the simple case of a sequence of a bi-level polygon, moving by translation, in a uniform background as illustrated in Figure 1. Clearly, the object has only two degrees of freedom. Intuitively, if we can accurately retrieve the barycenter or the centre of mass of each frame, the translation vector of the object between the two frames can be obtained. Let us define a video sequence of N frames to be $f_i(x,y)$, $i = 1,2,...,N$, $x,y \in \mathbb{R}$. The barycenter of each frame, $f_i(x,y)$, is defined by $(\overline{x}_i, \overline{y}_i) = \left(\frac{m_{1,0}}{m_{0,0}}, \frac{m_{0,1}}{m_{0,0}}\right)_i$ where $m_{p,q}$ is the continuous geometric moment as given in equation (3). The translation vector $\underline{t_i}$ between any two consecutive frames $f_i(x,y)$ and $f_{i-1}(x,y)$ is therefore $(t_{xi}, t_{yi}) = (\overline{x}_i, \overline{y}_i) - (\overline{x}_{i-1}, \overline{y}_{i-1})$.

Our coding scheme for this sequence is as follow; the first frame $f_1(x,y)$ is set to be the "key frame" and the location of each corner point of the polygon is quantized and transmitted to the decoder. The "non-key frames" $f_i(x,y)$, $i = 2,3,...,N$ are then sampled with a sampling kernel $\varphi(x,y)$ that satisfies the condition given in (2). The samples $S_i(m,n)$, $i = 2,3,...,N$ are quantized before being transmitted. The decoder calculates a set of barycenter $(\overline{x}_i, \overline{y}_i)$, $i = 1,2,...,N$ using the equations (3) and (4). The corresponding translation vectors $\underline{t_i}$ are then retrieved. Knowing the locations of each corner point of the key frame and the vectors $\underline{t_i}$, $i = 1,2,...,N$ the entire sequence can be reconstructed.

Since the geometric moments are retrieved from the samples $S_i(m,n)$ the distortion in our scheme is due to the quantization of these samples. In order to measure the compression efficiency, we compare the performance of our coding scheme with the theoretical distortion-rate $D(R_{total})$ upper bound of an ideal interframe coder, where the encoder only transmits the quantized corner points and the quantized translation vectors $\underline{t_i}$. The distortion D is measured as the mean-squared-error (MSE). We now briefly show the derivation of the distortion-rate $D(R_{total})$ bound of this ideal interframe coder.

Given an N-frame video sequence $f_i(x,y), i = 1,2,...,N$ with size $M \times M$ that contains a bi-level equal-side polygon of C corner points with amplitude B, there are two sets of parameter that the interframe encoder has to quantize: the corner points and the translation vectors. Since the quantization errors in each set of parameter are independent, the distortion can be separated into two types, namely, the distortion induced by the quantization of corner points, D_C, and the distortion induced by the quantization of the translation vector, D_T. We can therefore assume that D_C and D_T are additive and, thus, the total distortion for the entire sequence is given by:

$$D_{total} = ND_C + (N-1)D_T. \tag{5}$$

Furthermore, at high bit rate, we can also assume that, for each parameter, the quantization errors along x-direction and y-direction in the Cartesian coordinate are independent and, hence, also additive.

Using the above assumptions, at high bit rate, we can show that D_C and D_T are bounded by:

$$\begin{aligned} D_C &\leq 2M^2B^2(2^{-\frac{R_C}{2}}), & (6)\\ D_T &\leq 4M^2B^2(2^{-\frac{R_T}{2}}), & (7) \end{aligned}$$

where R_C and R_T are the number of bits allocated to represent each corner point and each translation vector respectively. Therefore, by minimizing D_{total} with respect to R_C and R_T, we can obtain the following optimal bit allocation:

$$R_C = R_T + R_E, \text{ where } R_E = 2\log_2\left(\frac{N}{2(N-1)}\right). \tag{8}$$

Clearly, the total number of bits used to encode the sequence is given by $R_{total} = CR_C + (N-1)R_T$. It then follows that by using the bit allocation equation in (8) and substituting (6) and (7) into (5), the theoretical distortion-rate $D(R_{total})$ curve of the interframe coder, where D is measured as the MSE, is bounded by:

$$\begin{aligned} D(R_{total}) &\leq 2M^2B^2E\left(2^{-\frac{R_{total}}{2(C+N-1)}}\right), \\ E &= 2^{\frac{CR_E}{2(C+N-1)}}\left(2^{-\frac{R_E}{2}} + \frac{2(N-1)}{N}\right). \end{aligned} \tag{9}$$

This $D(R_{total})$ curve will be used to examine how close the performance of our coding scheme is to the ideal interframe coder.

3.2. A real object moving under affine transform

We can extend the above scheme to the case of a real object, where its motion is estimated with an affine transform. The disparity between the two frames is given by $(x_i, y_i) = A(x_{i-1}, y_{i-1}) + \underline{t}$ where A is a non-singular 2×2 matrix and $\underline{t}$ is a translation vector. A method to retrieve the transformation matrix A using second and higher order moments is described in[7] and.[8] In,[7] the author showed that, by using the whitening transform, the estimation of A can be reduced to a problem of finding a rotational matrix R with:

$$A = F_g R F_f^{-1} \text{ with } F_{(\cdot)} = \begin{bmatrix} \sqrt{\mu_{2,0}^{(\cdot)}} & 0 \\ \frac{\mu_{1,1}^{(\cdot)}}{\sqrt{\mu_{2,0}^{(\cdot)}}} & \sqrt{\mu_{0,2}^{(\cdot)} - \frac{\mu_{1,1}^{(\cdot)2}}{\mu_{2,0}^{(\cdot)}}} \end{bmatrix} \tag{10}$$

where $\mu_{p,q}^{(\cdot)}$ is the central moment of order $(p+q)$. It was shown in[7] that the matrix R can be retrieved from the third order complex moments. Both the central moments and complex moments can be calculated from a combination of continuous geometric moments of the same order. Therefore, we need a sampling kernel that can reproduce polynomials up to degree three, for example, a third order B-Spline or a Daubechies db 4 in order to estimate such moments.

Since the key frame contains a real object, a conventional intraframe coding can be used. We used JPEG2000 to encode the first frame. The above scheme can then be repeated where the encoder only transmits the JPEG2000 encoded key frame and the low resolution samples of the non-key frames. The decoder retrieves A and $\underline{t}$ using the moments obtained by the equation (3) and reconstruct the sequence accordingly.

Finally, since our sampling kernel $\varphi_j(x, y)$ is, in fact, a scaling function, the decoder can apply the DWT to the reconstructed frame and replace the low-pass coefficients (at the j^{th} decomposition level) with the real, quantized, samples $\widehat{S}_{m,n}$. Inverse discrete wavelet transform (IDWT) can then be carried out to obtain a new image. This technique has the advantage of increasing the overall robustness of the scheme as a real set of data is used in the reconstruction process.

3.3. A real video sequence with fixed background

This is a part of our on-going work where we aim to apply the above framework to compress a video sequence with a fixed background. A good example is given by the "highway sequence*" as shown in Figure 7. From our observation, the motion of each car mainly involves translation and rescaling, which can be modelled by an affine transform. In this case, we assume that the encoder and decoder have access to the background image B_g by extracting it from a set of video frames prior to the compression process.

*Courtesy of the ACTS Project AC304 MODEST

We propose the following coding algorithm for this sequence. Firstly, the video frames are divided into key frames and non-key frames. The key frames K are encoded using a conventional intraframe coding method. We used JPEG2000 in our work. For the non-key frames L, the residual or the frame difference D_{res} between the current frame and the background is calculated. This frame difference is then sampled and the samples $S_{m,n}$ are fed into a quantizer. The encoder only transmits the quantized samples $\widehat{S}_{m,n}$ and the encoded key frames $K_{jpeg2000}$ to the decoder.

At the decoder, the key frame is reconstructed first. Object segmentation and separation are then performed on the reconstructed key frame K' and the objects O_o are obtained where $o = 1, 2, .., N_o$ and N_o denotes the total number of objects. The non-key frames are reconstructed as follow: first, the decoder segments the observed samples $\widehat{S}_{m,n}$ and combines the result with the sampled version of the background $B_{m,n}$ in order to obtain the sampled version of each object $\widehat{S}^{obj}_{m,n}$; for each object, the moments are retrieved using (3) and the corresponding affine transform matrix A_o is calculated; finally, the decoder combines the motion compensated objects O'_o with the background.

In order to improve the quality of the reconstructed non-key frames, an iterative reconstruction technique can be applied at this stage. The decoder, for instance, can re-sample the motion compensated object and compare the result with the "actual" samples $\widehat{S}^{obj}_{m,n}$. This process allows the affine transform parameters to be refined. Once again, the decoder can apply the DWT to the reconstructed frame and replace the low-pass coefficients with the real, quantized, samples $\widehat{S}_{m,n}$ plus the sampled version of the background and perform the IDWT to obtain a new image. The proposed encoding and decoding algorithm is summarised in the Figure 2. From the figure, we can see that the complexity of the encoder is much lower than that of the decoder.

Since the encoding process is very simple, the format of the output bit stream is much lighter than that of an interframe encoder. Therefore, there is much room for innovation in the decoding process. We are currently investigating the use of a more powerful decoder, which employs more sophisticated segmentation and interpolation techniques that will give more visually pleasant results.

4. QUANTIZATION OF SAMPLES

In our simulation, we adopted the quantization technique used in the embedded coding of images with zerotrees of wavelet coefficients as presented in.[9] The algorithm in[9] proceeds as follow: the DWT is first applied to the image; the encoder scans the image in order of importance determined by the precision, magnitude, scale and location of the wavelet coefficients; finally, a bit plane is generated and is fed into an arithmetic coder for further compression. The key point here is that larger coefficients are deemed more important by the encoder. We refer the reader to[9] for an excellent in-depth explanation of this method. Even though we only encode the low-pass coefficients and, therefore, do not have any zerotree in our scheme, the embedded coding still gives us the advantage of allocating more bits to the samples with higher magnitudes and provides a compact multiprecision representation of the samples. This method can provide a good result especially when the IDWT is applied to the quantized samples in the reconstruction phase.

One of the key steps in our proposed coding schemes is the retrieval of geometric moments from the low-resolution sampled version of the video frame. However, since the samples are quantized, errors are introduced. Let the quantized samples be $\widehat{S}_{m,n} = S_{m,n} + e_{m,n}$, where $e_{m,n}$ is the quantization error. It follows that, from equation (3), the errors in the retrieved moments $w_{p,q}$ are given by:

$$w_{p,q} = \sum_{m\in\mathbb{Z}} \sum_{n\in\mathbb{Z}} c^{(p,q)}_{m,n} e_{m,n}. \tag{11}$$

Therefore, given a limited bit budget, if the moments are to be retrieved as precisely as possible then more bits should be allocated to represent the samples that reduce $w_{p,q}$ the most. This means that we should allocate more bits to the samples $S_{m,n}$ whose corresponding coefficients $c^{(p,q)}_{m,n}$ are larger. The above quantization strategy is still sub-optimal in this case because it does not take into account the value of the coefficient. It is worth noting that, since the samples $\widehat{S}_{m,n}$ are also used in the IDWT, a quantizer that minimizes $w_{p,q}$ may not, however, provide the best reconstructed result.

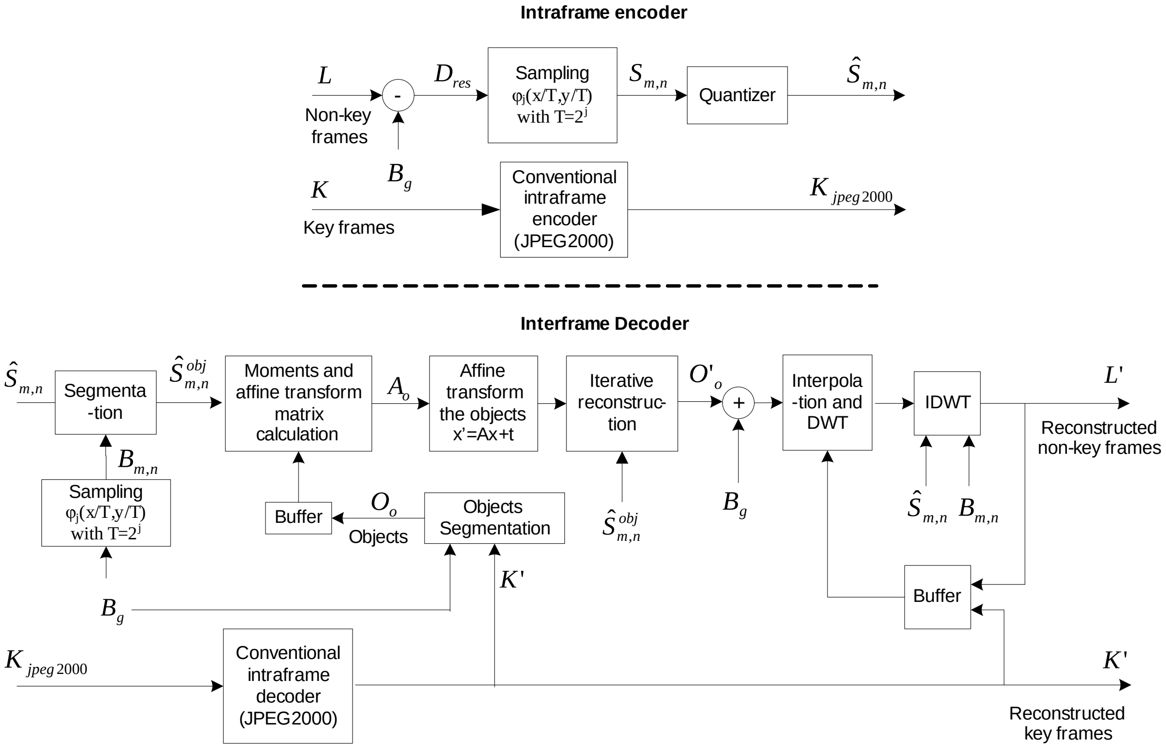

Figure 2. A coding scheme for the highway sequence. Top: the intraframe encoder. Bottom: the interframe decoder, which performs joint decoding.

Following from the above analysis, we therefore need to decide a suitable balance between the accuracy of the retrieved moments and the accuracy of the IDWT by choosing an appropriate quantization strategy. In this paper, we have chosen to use the embedded coding technique presented in.[9]

5. PRELIMINARY RESULTS

In this section, we present our preliminary results of the coding schemes for the three scenarios above, which will help demonstrating the concepts of this new approach.

5.1. A bi-level polygon moving by translation

First, we present the simulation results for the polygon case. The resolution of the original image was 1024×1024, which was then sampled with a Daubechies 2 filter for 8 iterations. The observed samples were of size 8×8. Note that zero-padding was used to eliminate errors caused by boundary conditions. The object was a bi-level square, which was then translated to form a sequence of 8 frames. We used 10 bits to encode the corner points and then varied the number of bits used by the embedded-code quantizer to represent each sample. The $D(R)$ plot is shown in Figure 3 (log scale) where the theoretical $D(R)$ of the interframe encoder is given in (9) with $M = 1$ and $B = 255$ in this case. At higher rate, the gap between the ideal encoder and our scheme is in the order of 10^{-4} bits per pixel (bpp). A perfect reconstruction was achieved at a total rate of 1.76×10^{-4} bpp, which is shown as the steep drop in the MSE value. Figure 4 shows the plot of the average percentage error in the retrieved first order moments against bit rate in log scale. When the error reached 0.133%, we obtained a perfect reconstruction of the sequence.

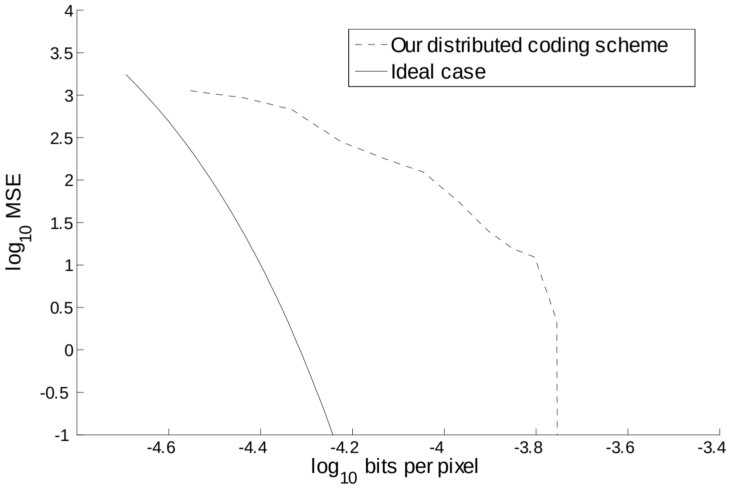

Figure 3. Plot of $D(R)$ (measured as MSE, in log scale) against bit rate used to encode the 8-frame polygon sequence: (Left) the ideal interframe encoder case (Right) our distributed coding scheme. A perfect reconstruction was achieved at 1.76×10^{-4} bpp.

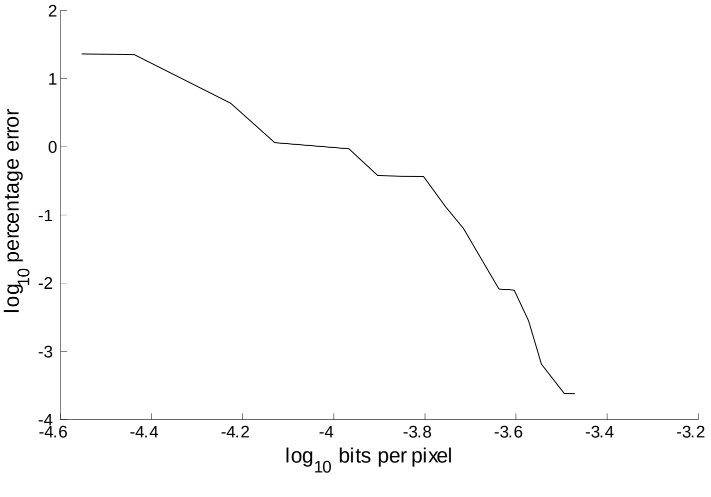

Figure 4. Plot of the average percentage error in the retrieved first order moments against bit rate in log scale. A perfect reconstruction was achieved when the error was at 0.133%.

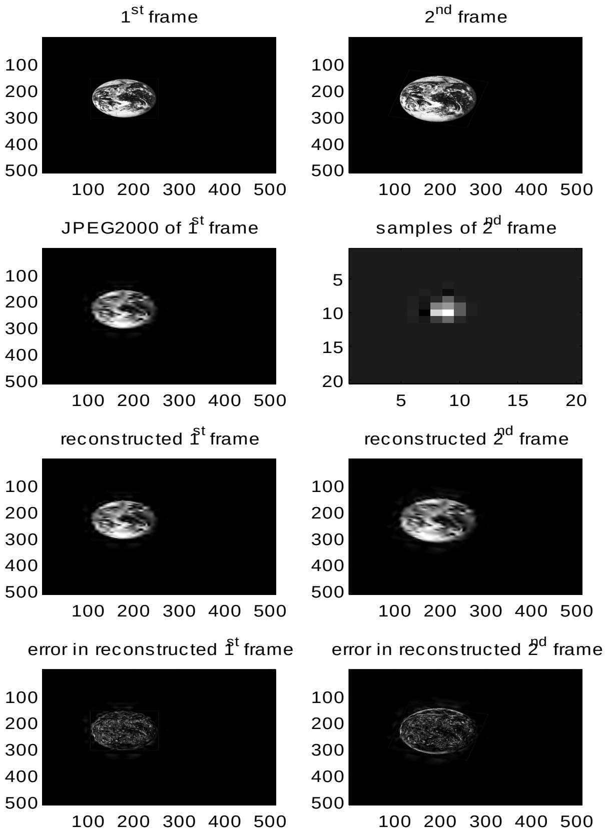

Figure 5. Encoding of a real object under affine transform, showing the key frame and the 2^{nd} frame: (top row) original sequence; (second row) transmitted JPEG2000 and samples; (third row) reconstructed key frame and 2^{nd} frame; (bottom row) corresponding errors in reconstruction.

5.2. A real object moving under affine transform

Figure 5 shows the simulation results of a real object moving under affine transform. The sequence had 8 frames, each of size 512×512. In this example, the object was translated, rotated and re-scaled. Each non-key frame was sampled with a Daubechies db 4 filter for 6 iterations. With zero-padding, the observed samples were 20×20 in size. The JPEG2000 encoding of the key frame was implemented with the JJ2000 software.[10] A plot of PSNR against bit rate, when the sequence as illustrated in Figure 5 was encoded with our proposed coding algorithm, is shown in Figure 6. The bit allocation between the key frame and the non-key frames was done using a greedy strategy, meaning that an additional bit was given to the one that improves PSNR the most. We compared our results with that of the, independent, JPEG2000 intraframe encoder where each frame was independently encoded with JPEG2000. From Figure 6, we can see that at lower rates (below the 0.01 bpp point), our coding scheme performed better than the JPEG2000 intraframe encoder.

5.3. A real video sequence with fixed background

Figure 7 shows the experimental results of the coding scheme for the highway sequence. The segmentation was done by simply detecting the changes in pixel values. The original frame was 288×352 and the sampled frames were 88×104. Multiple objects limited the number of sampling iteration in this case as we needed to separate

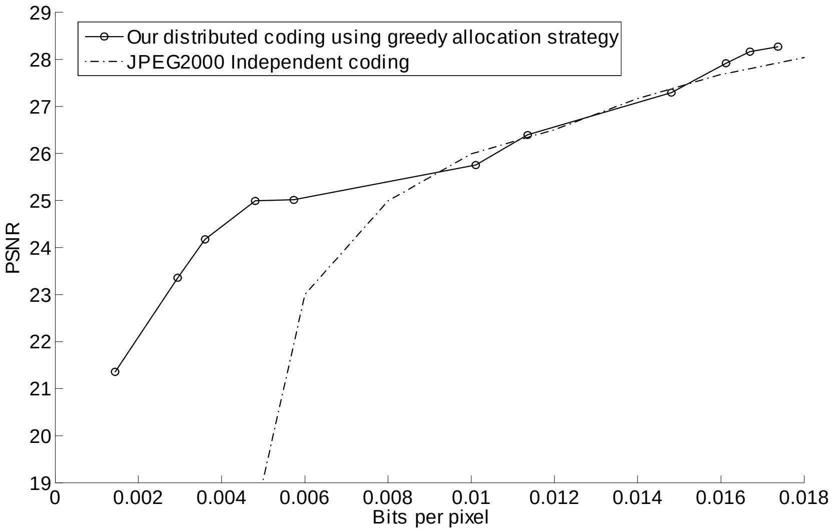

Figure 6. Plots of PSNR against bit rate for our proposed coding scheme and the JPEG2000 intraframe encoder. At lower rates (below 0.01 bpp), our coding scheme performed better than the JPEG2000 intraframe encoder.

the samples at the decoder. The affine transformation matrix was limited to rescaling and translation only. JPEG2000 was used to encode the key frame at 0.5 bpp. For a 4-frame sequence, we achieved PSNR of 37.7 dB at a bit rate of 0.1776 bpp. It is worth noting that the bit rate of the non-key frames was only 0.07 bpp. On the other hand, an intraframe, independent, JPEG2000 encoding at 0.1776 bpp would give a PSNR of 34 dB, therefore, our coding scheme achieved a better performance at this given rate.

From Figure 7, we can see that there were some ghosting effects in the area around the cars in the reconstructed second frame. This is because there was a miss-match between the real set of samples and the low-pass DWT coefficients that was replaced by the decoder. However, with the use of more sophisticated segmentation and interpolation techniques together with the iterative refinement of the affine transform parameters, we expect the visual quality to improve significantly. This development is a part of our on-going work.

6. CONCLUSIONS AND FUTURE WORK

We have introduced a new approach to a distributed coding problem using sampling of FRI signal. The novelty of the scheme is the ability to preserve and retrieve the geometric moments from the extremely low-resolution set of samples. Three schemes were presented in this paper. Our preliminary results show that for a sequence with real object and the highway sequence, our proposed coding schemes can outperform an independent JPEG2000 intraframe encoder at lower bit rate. Note, however, that these coding schemes are still in the early stage of development.

Our future work includes finding appropriate segmentation and object separation techniques as well as exploring various interpolation methods for the reconstruction of the non key frames. A more precise analysis on the effect of quantization errors and the $D(R)$ behavior of our schemes will be examined. A more optimal quantizer design and the bit allocation problem between the key frames and the samples will also be studied in greater detail. Finally, we aim to develop a scheme which employs more local motion estimation techniques at the decoder.

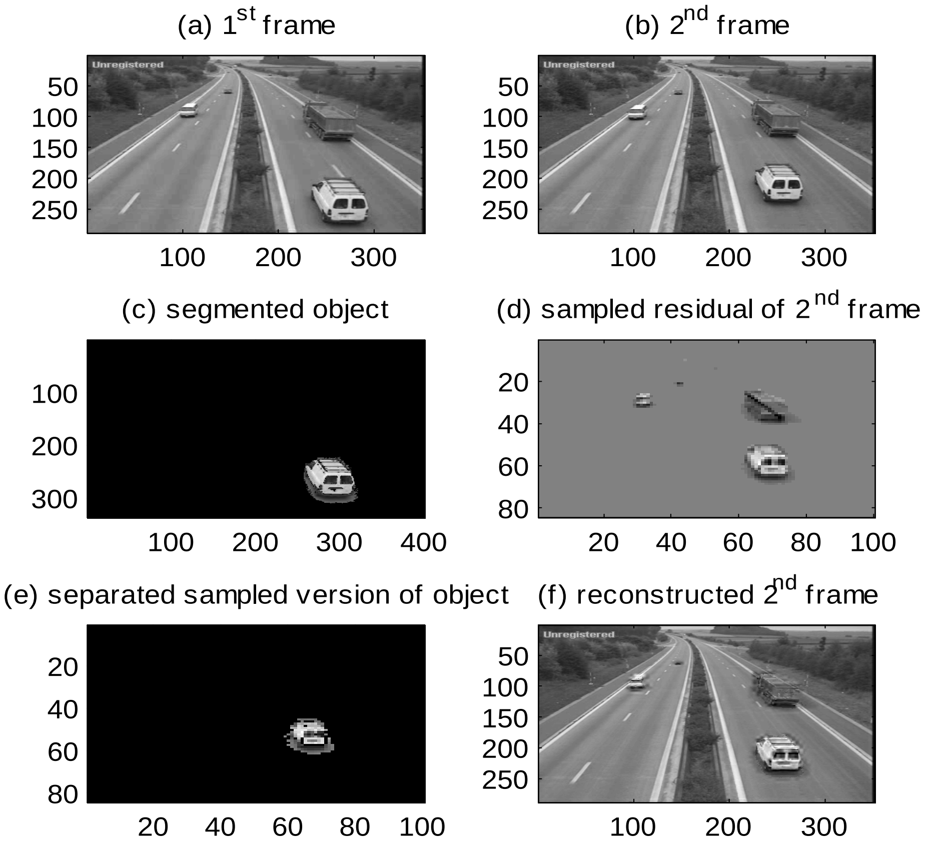

Figure 7. Simulation of the coding scheme for the highway sequence: (a) 1^{st} frame (b) 2^{nd} frame (c) segmented object at the decoder (d) sampled frame different of 2^{nd} frame (e) separated samples (f) reconstructed 2^{nd} frame

REFERENCES

1. P. Dragotti, M. Vetterli, and T. Blu, "Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang-Fix," *IEEE Transactions on Signal Processing* **55(5)**, pp. 1741–1757, May 2007.
2. B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE* **93**(1), pp. 71–83, 2005.
3. R. Puri and K. Ramchandran, "PRISM: a video coding architecture based on distributed compression principles," in *40th Allerton Conference on Communication, Controll and Computing*, (Allerton, IL), October 2002.
4. D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory* **19**(4), pp. 471–480, 1973.
5. A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory* **22**(1), pp. 1–10, 1976.
6. M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Transactions on Signal Processing* **50**(6), pp. 1417–1428, 2002.
7. J. Heikkilä, "Pattern matching with affine moment descriptors," *Pattern Recognition* **37**(9), pp. 1825–1834, 2004.
8. L. Baboulaz and P. Dragotti, "Distributed acquisition and image super-resolution based on continuous moments from samples," in *IEEE International Conference on Image Processing (ICIP)*, (Atlanta, USA), October 2006.
9. J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]* **41**(12), pp. 3445–3462, 1993.
10. "JJ2000 an implementation of the JPEG2000 standard in JavaTM." EPFL (Ecole Polytechnique Federale de Laussane), Switzerland. http://jj2000.epfl.ch/.