A NEW APPROACH TO DISTRIBUTED CODING USING 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, UK
{varit.chaisinthop02,p.dragotti}@imperial.ac.uk

ABSTRACT

This paper proposes a new approach to distributed video coding. Distributed video coding is based on the concept of decoding with side information at the decoder. Such a coding scheme employs a low-complexity encoder and the load of computational complexity is shifted to the decoder side. This property makes it well suited for low-power devices such as mobile video cameras.

The uniqueness of our approach lies in the combined use of discrete wavelet transform (DWT) and the concept of sampling of signals with finite rate of innovation (FRI) [1], which allow us to shift the task of motion estimation to the decoder side. Unlike the currently existing practical coders, we do not employ any traditional channel coding technique. Our preliminary results show that, for a simple video sequence with a uniform background, 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]. Such video coding schemes employ a low-complexity encoder while achieving 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. Sampling of FRI signals shows that the geometric moments of the signals can be retrieved from its low-resolution set of samples, which are the low-pass coefficients of the DWT. The motion parameters describing the disparity between two video frames can then be estimated using the moments of each frame. In our scheme, the encoder only performs the DWT and the complex task of motion estimation is shifted to the decoder side. Therefore, our aim here is to present a new approach to perform 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.

In the next section, main results of sampling of FRI signals are discussed. Our proposed distributed coding schemes are presented in Section 3. The preliminary results are given in Section 4. Finally, conclusions are drawn in Section 5.

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). The definition and sampling schemes of FRI signals are given in details in [4] and [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 $p,q \in \mathbb{Z}$ 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:

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

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 can therefore 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 level of the wavelet decomposition. Here, $\varphi_j(x,y)$ is given by the tensor product of two 1-D scaling function $\varphi_{j,n}(t) = 2^{-j/2}\varphi(2^{-j}t - n)$, $j,n \in \mathbb{Z}$. Thus we can change the resolution of $S_{m,n}$ by altering j with the corresponding sampling period of $T = 2^j$. It can be shown that the required coefficients $c_{m,n}^{(p,q)}$ are given by:

$$c_{m,n}^{(p,q)} = \left\langle x^p y^q, \widetilde{\varphi}_j(x-m, y-n) \right\rangle \tag{4}$$

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 the equation (3).

3. A NOVEL APPROACH TO DISTRIBUTED VIDEO CODING

The basic framework behind our coding scheme is as follows; first, the video sequence is divided into blocks where each block contains N frames. The first frame of each block is treated as the "key frame" and is encoded with a conventional intraframe coding method. The rest of the frames are "non-key frames", which are sampled and then quantized. At the decoder, once the key frame is reconstructed, its moments can be calculated directly. The moments of the non-key frames are retrieved from the quantized samples using the relationship given in (3). The motion parameters describing the disparity between the key frame and non key frames are then estimated using their respective moments. Lastly, non-key frames are reconstructed by performing motion estimation on the key frame. Compression is achieved by only transmitting the low-resolution set of samples of non-key frames. We now present our proposed distributed coding schemes for the three scenarios stated above.

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. Let us define a block of N video frames to be $f_i(x,y), i = 1,2,...,N, x,y \in \mathbb{R}$ and set $f_1(x,y)$ to be the key frame. It follows that $f_i(x,y) = f_1(x-x_i, y-y_i), i = 2,3,...,N$ where $\underline{t_i} = [x_i, y_i]$ is the translation vector. Using the definition of the moment of the i^{th} frame, $m_{p,q}^i$, and let $x' = x - x_i$, we have that:

$$\begin{aligned} m_{1,0}^i &= \iint f_1(x',y')(x'+x_i)dx'dy' \\ &= m_{1,0}^1 + m_{0,0}^1 x_i. \end{aligned} \tag{5}$$

Therefore $\underline{t_i}$ can be calculated using first and zeroth order moments as $x_i = \frac{(m_{1,0}^i - m_{1,0}^1)}{m_{0,0}^1}$ and similarly $y_i = \frac{(m_{0,1}^i - m_{0,1}^1)}{m_{0,0}^1}$. Note that $(\overline{x}, \overline{y}) = \left(\frac{m_{1,0}}{m_{0,0}}, \frac{m_{0,1}}{m_{0,0}}\right)$ is defined as the barycenter of the signal.

Our coding scheme for this sequence is as follow; each corner point of the polygon in the key frame $f_1(x,y)$ is quantized and transmitted to the decoder and the moments of $f_1(x,y)$ are computed directly. The non-key frames $f_i(x,y)$, $i = 2,3,...,N$ are then sampled with a 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 retrieves the zeroth and first order moments of each frame using the equations (3) and (4) and the translation vectors $\underline{t_i}$, $i = 2,3,...,N$ are retrieved as shown in (5). Non key frames are then reconstructed as $f_i(x,y) = f_1(x-x_i, y-y_i)$. This scheme is illustrated in Figure 1.

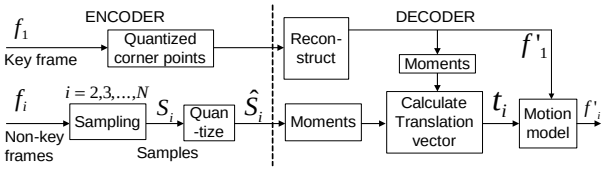

Fig. 1. Coding scheme for a translating bi-level polygon

Since the continuous moments are preserved in the samples $S_i(m,n)$ the distortion in our scheme is due to the quantization process. We compare the performance of our coding scheme with an ideal interframe coder case, where the encoder transmits the quantized corner points and the quantized translation vectors $\underline{t_i}$ directly to the decoder. Given a block of N frames, each of size $M \times M$, that contains a bi-level equal-side polygon of C corner points with amplitude B. At high bit-rate, we can show that the theoretical distortion-rate $D(R_{total})$ curve of the interframe coder, where D is measured as the mean-squared-error (MSE), is bounded by:

$$\begin{aligned} D(R_{total}) &\leq 2M^2B^2E\left(2^{-\frac{R_{total}}{2(C+N-1)}}\right), \\ E &= 2^{\frac{CR_E}{2(C+N-1)}}\left(2^{-\frac{R_E}{2}} + \frac{2(N-1)}{N}\right) \end{aligned} \tag{6}$$

with the constant $R_E = 2\log_2\left(\frac{N}{2(N-1)}\right)$. The total number of bits used to encode the sequence is given by $R_{total} = CR_C + (N-1)R_T$, where R_C and R_T are the number of bits allocated to represent each corner point and each translation vector in the Cartesian coordinate. The optimal bit-allocation is given by $R_T + R_E = R_C$. The derivation of the $D(R)$ curve is omitted due to limited space. The simulation results are given in Section 4.

3.2. A real object moving under affine transform

We can extend the above scheme to the case of a real object, where the motion is estimated with an affine transform. The disparity between the i^{th} and j^{th} frame is given by $(x_j, y_j) = A_{ij}(x_i, y_i) + \underline{t_{ij}}$, $i,j \in [1,2,...,N]$, $i \neq j$ where A_{ij} is a non-singular 2×2 matrix and $\underline{t}$ is a translation vector. A method to retrieve the matrix A_{ij} using second and higher order moments is described in [5] and [6]. In [5], the author showed that, by using the whitening transform, the estimation of A_{ij} can be reduced to a problem of finding a rotational matrix R with:

$$A = F_j R F_i^{-1} \text{ with } F_{(\cdot)} = \begin{bmatrix} \sqrt{\mu_{2,0}^{(\cdot)}} & 0 \\ \frac{\mu_{1,1}^{(\cdot)}}{\sqrt{\mu_{2,0}^{(\cdot)}}} & \sqrt{\mu_{0,2}^{(\cdot)} - \frac{\mu_{1,1}^{(\cdot)2}}{\mu_{2,0}^{(\cdot)}}} \end{bmatrix} \tag{7}$$

where $\mu_{p,q}^{(\cdot)}$ is the central moment of order $(p+q)$. It was shown in [5] that the matrix R can be retrieved from the third order complex moments. The central and complex moments can be calculated from a combination of 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.

The above coding scheme, as shown in Figure 1, can be repeated in a similar manner. Since the sequence contains a real object, the key frame can be encoded with a conventional intraframe

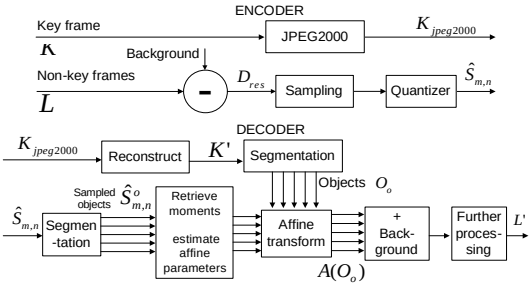

Fig. 2. A coding scheme for the highway sequence. Top: the encoder. Bottom: the decoder.

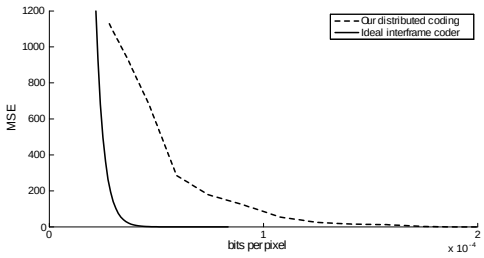

Fig. 3. Plot of $D(R)$ (measured as MSE) against the bit rate for the polygon sequence: (Left) ideal interframe encoder (Right) our distributed coding scheme.

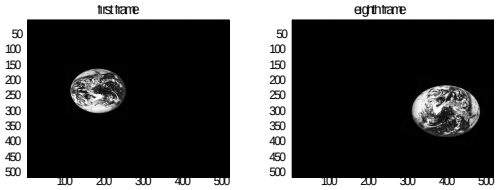

Fig. 4. A video sequence containing a real object undergoing affine transform. (Left) first frame (Right) eighth frame.

coding such as JPEG2000 and its moments can be computed directly at the decoder. Using the quantized samples $\widehat{S}_i(m,n)$, $i = 2,3,...,N$ of the non-key frames, the decoder estimates not only $\underline{t_{ij}}$ but also A_{ij} using the moments up to third order obtained by the equations (3) and (7). The sequence is reconstructed as $(x_j, y_j) = A_{ij}(x_i, y_i) + \underline{t_{ij}}$.

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 encode a more realistic video sequence with a fixed background. We used the "highway sequence" in our experiment as shown in Figure 6. The motion of each car can be modelled by an affine transform, which involves translation and rescaling. We also assume that the encoder and decoder have access to the background image by extracting it from a set of video frames prior to compression.

We propose the following coding algorithm for this sequence. First, the video sequence is divided into blocks of N frames. The first frame of the block is the key frame K and is encoded using a conventional intraframe coding method. For the non-key frames L, the frame difference D_{res} between the current frame and the background is sampled. The encoder then transmits these quantized samples $\widehat{S}_{m,n}$. At the decoder, object segmentation is performed on the reconstructed key frame K' and the objects O_o, $o = 1,2,..,N_o$ are obtained where N_o denotes the total number of objects or cars in this case. The moments of each object are then computed. The non-key frames are then reconstructed as follow: the decoder segments the observed samples to obtain the sampled version of each object $\widehat{S}^o_{m,n}$; for each object, the moments are retrieved and the affine transform parameters, A^o_{ij} and t^o, describing the disparity between the same object in the current frame and the key frame are estimated; finally, the decoder combines the affine transformed objects $A(O_o)$with the background.

In order to improve the quality of the reconstructed non-key frames, further processing can be applied, for example, iterative refinement of motion parameters using the observed samples as reference. The proposed coding scheme is summarised in the Figure 2. Note that the complexity of the encoder is much lower than that of the decoder. We are currently investigating the use more sophisticated segmentation and reconstruction techniques that will give more visually pleasant results. In the next section, we present our preliminary results of the coding schemes of the three scenarios above.

4. PRELIMINARY SIMULATION RESULTS

In order to quantize the samples in our simulation, we adopted the embedded coding technique first presented in [7]. This gave us the advantage of allocating more bits to samples with higher magnitudes and providing a compact multiprecision representation of samples.

First, we present the simulation results for the polygon case. The resolution of the original image was 1024×1024. It 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 sequence contained a translating bi-level square of 8 frames. The first frame was set to be the key frame. We used 10 bits to encode the corner points and then varied the number of bits used by the embedded-code quantizer to represent the samples. The $D(R)$ plot is shown in Figure 3 where the theoretical $D(R)$ of the interframe encoder is given in (6) 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.

Figure 4 shows a video sequence 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. JPEG2000 was used to encode the first frame and was implemented with the JJ2000 software [8]. A plot of PSNR against bit rate of the sequence as illustrated in Figure 4 is shown in Figure 5. 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 5, at lower rates (below the 0.01 bpp

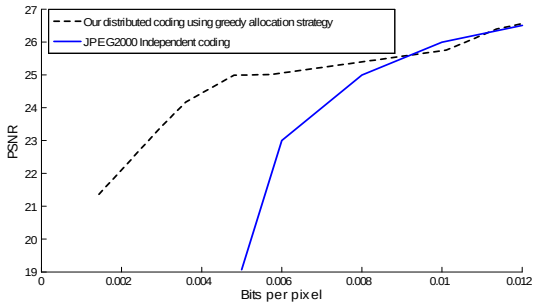

Fig. 5. Plots of PSNR against bit rate for the real object with affine transform case. At lower rates (below 0.01 bpp), our coding scheme performed better than the JPEG2000 intraframe encoder.

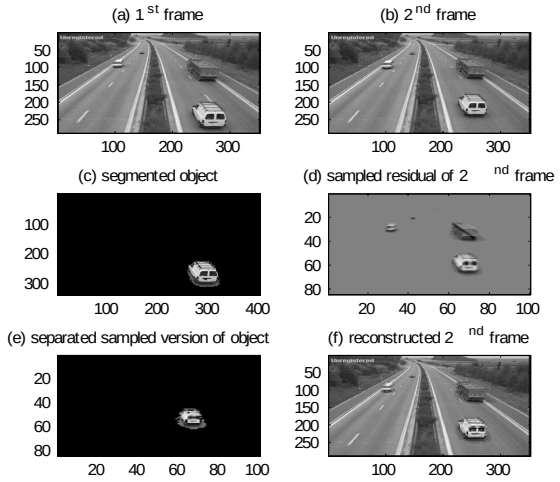

Fig. 6. Simulation of the coding scheme for the highway sequence: (a) 1^{st} frame (b) 2^{nd} frame (c) one of segmented objects at the decoder (d) sampled frame different of 2^{nd} frame (e) separated samples of one object (f) reconstructed 2^{nd} frame

point), our coding scheme performed better than the JPEG2000 intraframe encoder.

The highway sequence is shown in Figure 6. We used a 4-frame sequence with the original frame size of 288×352 and the samples were 88×104 with Daubechies db 4 filter at 2 decomposition level. Multiple objects limited the number of decomposition level as we needed to separate the samples at the decoder. JPEG2000 was used to encode the first frame. The segmentation was done by detecting changes in pixel values. Figure 7 shows the plot of PSNR against bit rate in comparison with a JPEG2000 intraframe encoder. In order to make a fair comparison, we assumed that the background is also available at the intraframe encoder. From Figure 7, the independent encoder performed slightly better as the performance of our scheme is limited by the simple segmentation technique. We believe that with more sophisticated segmentation and reconstruction techniques the performance can be improved significantly. This development is a part of our ongoing work.

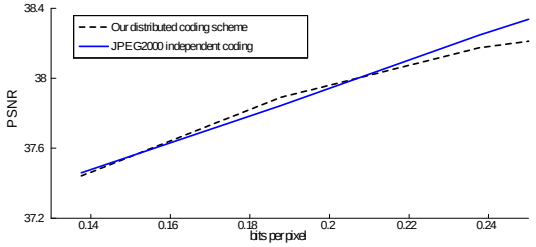

Fig. 7. Plots of PSNR against bit rate for the high way sequence.

5. CONCLUSIONS AND FUTURE WORK

In distributed video coding, the load of computational complexity is shifted to the decoder side. We have introduced a new approach to perform the complex task of motion estimation at the decoder using results of sampling of FRI signals. The novelty of the scheme lies in its ability to estimate the motion parameters using geometric moments, which can be retrieved from the low-resolution samples. Three schemes were presented in this paper. Our preliminary results showed that, for a simple video sequence with a uniform background, our coding scheme can perform better than an independent JPEG2000 intraframe coder at low bit rates. Our future work includes the development of a decoder that employs more sophisticated techniques as well as a more precise analysis on the effect of quantization errors and the $D(R)$ behavior of our schemes. Finally, we aim to develop a scheme which employs local motion estimation technique at the decoder.

6. REFERENCES

[1] P. L. 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*, vol. 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*, vol. 93, no. 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, Control and Computing*, Allerton, IL, October 2002.

[4] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *IEEE Transactions on Signal Processing*, vol. 50, no. 6, pp. 1417–1428, 2002.

[5] J. Heikkilä, "Pattern matching with affine moment descriptors," *Pattern Recognition*, vol. 37, no. 9, pp. 1825–1834, 2004.

[6] 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.

[7] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

[8] "JJ2000 an implementation of the JPEG2000 standard in JavaTM," EPFL (Ecole Polytechnique Federale de Lausanne), Switzerland, http://jj2000.epfl.ch/.