Proc. IEEE International Conference on Image Processing, Vol. 3, pp. 638-641, 1997.

Motion compensation of wavelet coefficients for very low bit rate video coding

François G. Meyer, [†] [⋆], Amir Averbuch [‡], and Ronald R. Coifman [†]
[†]Department of Mathematics, Yale University, 12 Hillhouse ave, New Haven, CT 06520, USA
[⋆]CNRS UMR-5515, Creatis, Insa 502, 69621 Villeurbanne, France
[‡]School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel
e-mail: francois.meyer@yale.edu

Abstract

We construct a new algorithm for motion compensation of wavelet coefficients. The new approach outperforms standard block matching techniques both in terms of higher PSNR, and better perceptual quality. The algorithm works directly in the wavelet coefficients domain and has been successfully used for wavelet based very low bit rate video coding.

1 Introduction

The recent emergence of multimedia applications has motivated a considerable interest in the development of video compression algorithms aimed at the transmission (or storage) of video sequences over channels with limited capacity. The complexity of the overall video signal may be efficiently reduced by decorralating the signal along the temporal dimension. This is usually achieved by estimating the motion between successive frames in order to "realign" each frame with respect to the previous one. The "realigned" video sequence can then be efficiently coded as the combination of some reference images and the differences between the reference images and the rest of the realigned frames. Existing video compression standards use 2D Discrete Cosine Transform (DCT) to code the reference image, and block matching in the image domain to realign the frames. Since the Wavelet Transform (WT) outperforms the DCT for the coding of still images, it is natural to build new video coding schemes that use a WT to compress the reference frames. It would be desirable to perform the motion estimation and the motion compensation directly on the wavelet coefficients. This paper addresses the problem of estimating the motion, and performing the motion compensation in the wavelet coefficients domain.

2 Motion compensation in the wavelet domain

Wavelet expansions are highly dependent on the alignment of the signal and the discrete grid chosen for the analysis. In order to perform the motion compensation in the wavelet domain, we need to be able to predict the coefficients of a signal that has been translated (more complex motions are decomposed into local translations). We consider a one dimensional signal $f[n]$, but the results are easily extended to the two-dimensional case. If the signal is shifted by an even number of samples $2p$, then the wavelet coefficients are shifted by p. Let us assume now that the signal has been shifted by an odd number of samples ; in fact we need only to understand the case where the shift is 1. Let s^1 be the lowpass wavelet coefficients of the original signal: $s^1[n]$ is the value of the undecimated lowpass coefficients of the unshifted signal at the grid point $2n$. Let s^1_t be the lowpass wavelet coefficients of the translated signal. Since $s^1_t[n]$ is the value of the undecimated lowpass filtered unshifted signal at $2n-1$, we can interpolate $s^1_t[n]$ from $s^1[n-1]$, $s^1[n]$ and $s^1[n+1]$. The lowpass filtered signal is smooth, and the interpolation error will be small. We conclude that it is possible to accurately interpolate the lowpass coefficients of the shifted signal from the lowpass coefficients of the unshifted signal.

Let d^1 be the highpass wavelet coefficients of the original signal, and let d^1_t be the highpass coefficients of the translated signal. As opposed to the lowpass coefficients, large highpass coefficients correspond to parts of the signal such as edges, or singularities. The highpass image will not be a smooth image, and it will contain singularities. Consequently it will not be possible to accurately interpolate the highpass coefficients of the shifted signal from the highpass coefficients of the unshifted signal. We can try to better understand this phenomena. We consider here the Haar wavelet, Ψ, to simplify the exposition. We have:

$$\begin{aligned} d^1[n] &= \tfrac{1}{\sqrt{2}}\left(f[2n+1]-f[2n]\right) \\ d^1_t[n] &= \tfrac{1}{\sqrt{2}}\left(f[2n]-f[2n-1]\right) \end{aligned} \qquad (1)$$

$d^1[n]$ will be large when there is an edge around $2n$ in the original signal. We consider two types of edges as show in Fig. 1.

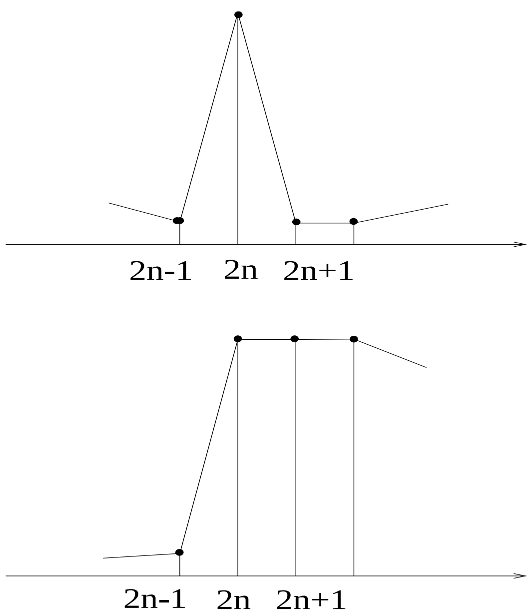

Figure 1: Two type of edges, top: “impulse” edge, bottom “ramp” edge.

- impulse function located at $2n$: We have $f[2n-1] = f[2n+1] = f[2n+2] = \varepsilon$, and $f[2n] = \lambda \gg \varepsilon$.

  Then $d^1[n] = -\lambda$, and $d_t^1[n] = \lambda$. The highpass coefficient of the translated signal has became the opposite of the coefficient of the original signal.

- ramp function: we have $f[2n-1] = \varepsilon$, $f[2n] = f[2n+1] = f[2n+2] = \lambda \gg \varepsilon$. In this case $d^1[n] = 0$, but $d_t^1[n] = \lambda$.

Such phenomena will happen frequently around the edges of the image. This shows that it is not possible to predict the large highpass wavelet coefficients.

Several authors [1,2,3,4] have proposed to directly predict the lowpass and highpass wavelet coefficients using motion compensation. In [1] the authors have also raised the problem that we have documented here. Instead of predicting directly the highpass coefficients at a given resolution, they perform the motion compensation on the lowpass image at the next finer resolution, and they calculate the decimated highpass coefficients of this predicted lowpass image. At the finest resolution, their scheme effectively requires the motion compensation of the entire image. We propose a different approach, where we preserve the advantage of the wavelet representation: sparse highpass coefficients, and coarse lowpass representation. We first define a new pyramid structure that will permit us to predict the highpass wavelet coefficients of a shifted image.

3 A new pyramid structure for motion compensation of wavelet coefficients

As opposed to a traditional wavelet expansion, we do not decimate the highpass coefficients at each resolution. In practice this is not a major overhead since many of these coefficients are set to zero after quantization when compressing at the rate required for very-low bit rate applications. The lowpass coefficients in our pyramidal representation also differ from the traditional wavelet lowpass coefficients: the signal is lowpass filtered twice, instead of only once. Let H be the highpass filter, let L be the lowpass filter, and let $2\downarrow$ be the decimation operator (subsampling by 2). Let I be a 2-D image. The algorithm that calculates the coefficients of the pyramid is defined as follows (see Fig. 2):

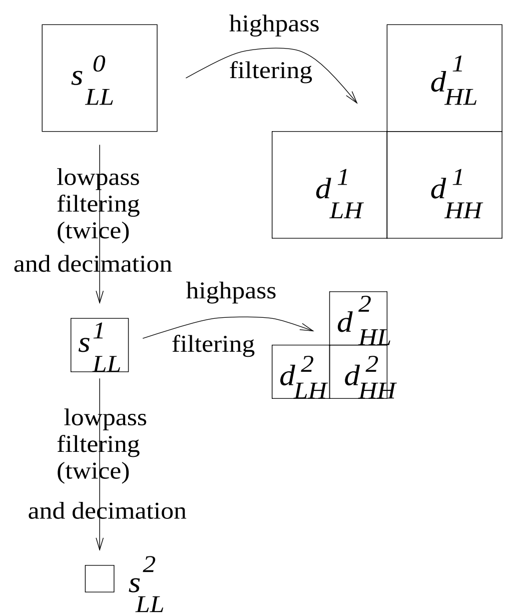

Figure 2: At each level we calculate the undecimated highpass coefficients, and the decimated lowpass coefficients

- Initialization at level 0. The lowpass component at the level 0 is equal to the original image:

$$s^0_{LL} = I. \tag{2}$$

- Decomposition at level $j+1$. Let s^j_{LL} be the lowpass signal calculated at level j.

  1. Calculate the undecimated highpass coefficients

$$\begin{aligned} d^{j+1}_{LH} &= LHs^j, \\ d^{j+1}_{HL} &= HLs^j, \\ d^{j+1}_{HH} &= HHs^j \end{aligned} \tag{3}$$

2. Calculate the decimated lowpass coefficients of the undecimated lowpass coefficients:

$$s^{j+1}_{LL} = 2\!\downarrow LL\, LL\, s^{j}_{LL} \tag{4}$$

s^{j+1}_{LL}, is not kept in the pyramid, but it will be the input of the decomposition at the next coarser scale.

- Final level J:
 1. Calculate the undecimated highpass coefficients: $d^J_{LH}, d^J_{HL}, d^J_{HH}$.
 2. Calculate the decimated lowpass filtered coefficients: s^J_{LL}, and save it into the pyramid.

3.1 Reconstruction of an image from the pyramid

Lemma 1 *We can reconstruct a 2-D image I from*

(i) the decimated twice lowpass filtered image

$$s^1_{LL} = 2\!\downarrow LL\, LL\, I$$

(ii) and the three undecimated highpass filtered images

$$\begin{aligned} d^1_{LH} &= LH\, I \\ d^j_{HL} &= HL\, I \\ d^j_{HH} &= HH\, I \end{aligned}$$

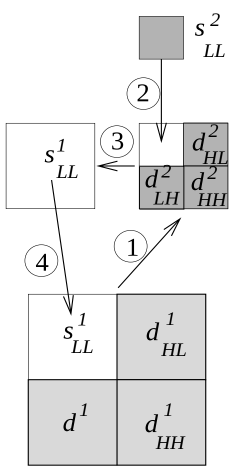

Figure 3: Reconstruction algorithm. (1) Lowpass filter and decimate the highpass coefficients d^j. (2) Add the lowpass coefficients s^j_{LL}, and (3) apply the inverse wavelet transform to recover the undecimated lowpass coefficients. (4) Put the undecimated lowpass coefficients back with the undecimated highpass coefficients.

The proof exploit two ingredients [5].

First we reconstruct the undecimated lowpass filtered image, $LL\, I$, from the decimated twice lowpass filtered

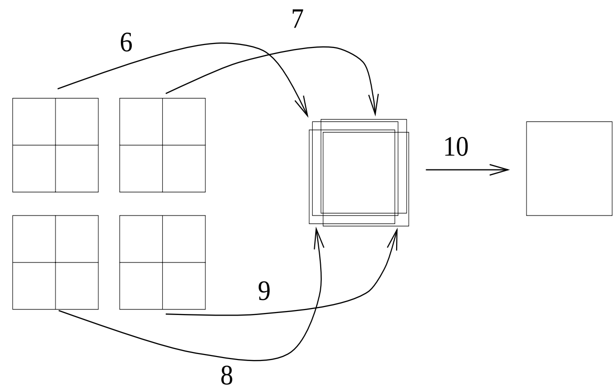

Figure 4: Reconstruction algorithm. (5) Extract four sets of decimated filtered coefficients and (6,7,8,9) reconstruct four shifted images. (10) Average the four images.

image, and from the undecimated highpass images, as shown in Fig. 3. We apply the lowpass filter LL on the undecimated highpass images: $d^1_{HL}, \ldots$ and we decimate. We notice that we can commute the highpass filters $HL, \ldots$ and the lowpasss filter LL before the subsampling. Finally we obtain

$$\begin{aligned} 2\!\downarrow LL\, d^1{}_{HL} &= 2\!\downarrow HL\,(LL\, I) \\ 2\!\downarrow LL\, d^1{}_{LH} &= 2\!\downarrow LH\,(LL\, I) \\ 2\!\downarrow LL\, d^1{}_{HH} &= 2\!\downarrow HH\,(LL\, I) \end{aligned} \tag{5}$$

Since we also have the decimated lowpass filtered signal, $2\!\downarrow LL\,(LL\, f)$, we have all the coefficients of the traditional wavelet transform of $LL\, I$. As a result we can reconstruct the lowpass filtered image $LL\, I$ with the traditional inverse wavelet transform.

At this point we have reconstructed the undecimated lowpass and undecimated highpass filtered images. Instead of decimating these filtered coefficients and applying an inverse wavelet transform, we will pursue a different approach that will result in a reconstruction of much better quality. We reconstruct four shifted images from the undecimated coefficients and we average them to obtain the final image, as shown in Fig. 4. There is a net advantage in averaging the four shifted reconstructed images over reconstructing only from the unshifted image: the reconstruction is smoother, and the reconstruction is more accurate (the order of the approximation is higher) [5,6]. The reconstruction algorithm starts at the coarsest level J, and reconstruct s^j_{LL} for each level $j = J, \ldots, 0$ using the lemma, until we reach the level 0.

4 Motion estimation and motion compensation

The motion between two frames is estimated with a hierarchical block matching approach. We tile the current image into blocks of same size, and we assume that the velocity is constant over a block. For each block we search the best match in the next image. We solve the problem with a hierarchical strategy. We exploit the pyramid of the lowpass wavelet coefficients for each frame. The number of blocks is kept constant at each resolution. An initial motion estimate is obtained with the coarsest resolution images. At each higher resolution we calculate an incremental value of the blocks displacements. The hierarchical approach also regularizes the motion field and is much faster than the brute force search at the finest resolution level.

The motion compensation of an image $I(t+1)$ from an image $I(t)$ is performed as follows. We first build the pyramid structure of image $I(t)$. We estimate the motion between $I(t)$ and $I(t+1)$. We then predict the pyramid of image $I(t+1)$ by performing the motion compensation of the coefficients at each level of the pyramid of $I(t)$. Motion compensation of the highpass images is not problematic, since we work here on the undecimated coefficients. Finally we can reconstruct the predicted image at $t+1$ from the predicted pyramid.

5 Experiments

We have implemented the wavelet based motion compensation algorithm as a part of a very low bit rate video coding algorithm [5]. We have compared the performance of the new algorithm against standard block matching techniques. Figure 5.a shows the result of the prediction of frame 51 from frame 50 of the QCIF test sequence *Susie* using integer resolution (8×8) block matching. We note the visually unpleasant blocking artifacts. Figure 5.b shows the result of the wavelet based motion compensation (8×8 blocks at the finer resolution). Even though there is a significant motion between the two frames, the new prediction has a better PSNR, and has a better perceptual quality. Also note that the edges (the highpass coefficients) have been accurately predicted.

6 References

[1] P. Cheng, J. Li, and C.-C J. Kuo. Multiscale video compression using wavelet transform and motion compensation. In *Intern. Conf. on Image Process., ICIP'95*, pages 606–609, 1995.

[2] A. Nosratinia and M.T. Orchard. Multi-resolution backward video coding. In *Intern. Conf. on Image Process., ICIP'95*, pages 563–566, 1995.

[3] K.H. Yang, S.J. Lee, and C.W. Lee. Motion-compensated wavelet transform coder for very low bit-rate visual telephony. *Signal Processing*, pages 581–592, 1995.

[4] Y-Q. Zhang and S. Zafar. Motion compensated wavelet transform coding for color video compression. *IEEE Trans. Circuits Syst. Video Technol.*, 2(3):285–296, 1992.

[5] F.G. Meyer, A. Averbuch, D. Lazar, and R.R. Coifman. Motion compensated wavelets for very-low bit rate video coding. Yale University, Department of Mathematics, Research report, 1997.

[6] R.R. Coifman and D. Donoho. Translation Invariant De-Noising. In *Wavelets in Statistics*, Antoniadis and Oppenheim Editors, Springer, 1995.

a

b

Figure 5: Sequence Susie. Reconstruction of frame 51 from frame 50. Left (a): block matching (8x8 blocks), PSNR = 32.39, Right (b): motion compensation of wavelet coefficients (8x8 blocks at the finer resolution), PSNR = 33.03.