MOTION COMPENSATION IN TEMPORAL DISCRETE WAVELET TRANSFORMS

Wei Zhao

August 2004

Boston University

Department of Electrical and Computer Engineering

Technical Report No. ECE-2004-04

BOSTON
UNIVERSITY

MOTION COMPENSATION IN TEMPORAL DISCRETE WAVELET TRANSFORMS

Wei Zhao

Boston University
Department of Electrical and Computer Engineering
8 Saint Mary's Street
Boston, MA 02215
`www.bu.edu/ece`

August 2004

Technical Report No. ECE-2004-04

Summary

Motion-compensated hybrid DPCM/DCT approach has dominated video coding over the last two decades. Best examples are successful MPEG and H.26X coders. Today, these coders are being challenged by new algorithms based on the discrete wavelet transform (DWT). Early approaches based on 2D (spatial) subband decomposition followed by motion compensation did not perform well primarily due to inefficiency of the motion-compensation of aliased signal components. However, various methods extending the wavelet transform to 3D (space-time) have shown great promise; works of Ohm, of Choi and Woods, and of Kim, Xiong and Pearlman have all exploited temporal correlation in 3D-DWT-transformed data. Although some methods exploited motion compensation to account for this correlation, the problem turned out to be challenging due to difficulties with temporal motion continuity. As an extension of these early results, recently several researchers have proposed separable 3D DWT (1D temporal transform followed by 2D spatial transform) with motion adaptation. Some of the methods proposed use transversal (standard) implementation of the motion-compensated temporal transform, while others use lifted implementation of the same transform. Although, in general, these implementations are equivalent, under motion-compensation this equivalence occurs only under certain conditions. Recently, Konrad has derived necessary and sufficient condition on such equivalence; the general condition states that motion composition must be a well-defined operator (for the case of Haar DWT the condition is that motion must be invertible). However, since not all motion models obey such properties (block matching is one example), in this report we investigate coding performance under different motion compensation scenarios. In particular, we compare the performance of two independently-estimated vector fields (forward and backward) with that of a single vector field (forward) plus interpolation of the backward vector field in both transversal and lifting implementations. Our experimental results show that the interpolation-based method outperforms the independent estimation in terms of coding PSNR not even considering the lower rate needed to transmit one single vector field instead of two.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The goal of video compression is to reduce the amount of data to represent a video sequence thus facilitating video transmission and storage. The main idea in video compression is to exploit spatial correlation between pixels in the same frame and temporal correlation between consecutive frames of a video sequence [6]. In the past two decades, motion-compensated hybrid DPCM/DCT video coding algorithms have been very successful in all kinds of applications. With the rapid development of new telecommunication services and electronic devices, people expect to receive video from different types of terminals and through channels with different capacity. Thus, scalability of video coding methods is highly desired and significant effort has been put into the design of scalable video coding algorithms.

In 2003, Secker and Taubman [3] proposed a new framework for highly scalable video compression called *lifting-based invertible motion adaptive transform* or LIMAT. In this framework, lifting-based implementation of the discrete wavelet transform (DWT) is applied along the temporal direction under motion compensation. This technique allows for very high temporal scalability [2] and also utilizes spatial scalability of JPEG-2000 image compression standard, while it is more efficient computationally than the transversal implementation. Recently, Konrad derived necessary and sufficient conditions on motion transformation for perfect reconstruction in motion-compensated transversal DWT, and also for the equivalence between motion-compensated transversal and lifted implementations of the temporal DWT [1]. In general, the condition is that the motion model must allow composition; a special case, e.g., for the Haar DWT, is that motion must be invertible. In practice, many motion models do not obey such properties. In this project, we investigate coding performance under different motion compensation scenarios. In particular, we compare the performance of two independently-estimated vector fields (forward and backward) with that of a single vector field (forward) plus interpolation of the backward vector field using a nearest-neighbor interpolator. We perform this comparison for both transversal and lifting implementations. Our experimental results show that the interpolation-based method outperforms the independent estimation in terms of coding PSNR not even considering the lower rate needed to transmit one single vector field. These results confirm the importance of motion invertibility in the case of the

Haar DWT, and thus suggest a new approach to motion compensation in compression methods based on temporal DWT.

In the next chapter, we review the necessary background material needed to understand details of the methods investigated. Chapter 3 describes the basic methods and algorithms used in experiments. Chapter 4 shows the results and discusses conclusions.

Chapter 2

Motion-compensated discrete wavelet transforms

2.1 Transversal and lifting approach to DWT

The discrete wavelet transform in temporal direction can be implemented using various kernels, such as Haar, 5/3, 9/7, etc. For each such kernel, there are two different ways of implementation: transversal approach (standard FIR filtering) and lifting-based approach. The lifting-based approach is more efficient computationally than the transversal approach. We can see this by using the 5/3 kernel as an example.

Let f_k denote the k-th frame of an image sequence, and let x denote spatial position of a pixel in this frame. The discrete wavelet transform based on the 5/3 filters can be described ed by the following equations:

- 5/3 transversal analysis equations:

$$\begin{aligned} h_k[x] &= f_{2k+1}[x] - \frac{1}{2}(f_{2k}[x] + f_{2k+2}[x]) \\ l_k[x] &= \frac{3}{4}f_{2k}[x] + \frac{1}{4}(f_{2k-1}[x] + f_{2k+1}[x]) - \frac{1}{8}(f_{2k-2}[x] + f_{2k+2}[x]) \end{aligned}$$

- 5/3 transversal synthesis equations:

$$\begin{aligned} f_{2k}[x] &= l_k - \frac{1}{4}(h_{k-1}[x] + h_k[x]) \\ f_{2k+1}[x] &= \frac{3}{4}h_k[x] - \frac{1}{8}(h_{k-1}[x] + h_{k+1}[x]) + \frac{1}{2}(l_k[x] + l_{k+1}[x]) \end{aligned}$$

- 5/3 analysis equations using lifting:

$$\begin{aligned} h_k[x] &= f_{2k+1}[x] - \frac{1}{2}(f_{2k}[x] + f_{2k+2}[x]) \\ l_k[x] &= f_{2k}[x] + \frac{1}{4}(h_{k-1}[x] + h_k[x]) \end{aligned}$$

- 5/3 synthesis equations using lifting:

$$\begin{aligned} f_{2k}[x] &= l_k[x] - \frac{1}{4}(h_{k-1}[x] + h_k[x]) \\ f_{2k+1}[x] &= h_k[x] + \frac{1}{2}(f_{2k}[x] + f_{2k+2}[x]) \end{aligned}$$

From the above equations we can see that in the lifting-based approach, the second equation (called the update step) always uses the result from the first equation (called the prediction step). Thus, it is more efficient computationally than the transversal approach and, therefore, more desirable in practical applications.

2.2 Temporal decomposition under motion compensation

By carrying out the temporal decomposition without motion compensation, however, significant energy will concentrate in the high subband (temporal prediction is inefficient without motion compensation). This is not desirable in compression. To achieve better compression performance, we need to incorporate motion information into the temporal DWT (Fig. 2.1). If the motion model works well, we can apply the temporal filters from previous section along motion trajectories of each sample position, and thus reduce energy in the high subband [3]. This is illustrated in Fig. 4.1.

Then, a question arises: Will the transversal and lifting-based approaches be still equivalent, and will they allow perfect reconstruction in presence of motion compensation? The answer is easy for the lifting approach since non-linearities introduced into the lifting equations do not prevent perfect reconstruction. However, this is not clear for the transversal approach. Also, equivalence between the two approaches is not a given. Recently, Konrad has derived the necessary and sufficient conditions for perfect perfect reconstruction of the transversal implementation and also for the equivalence of the two approaches under motion compensation. This condition states that motion transformation must allow composition [1]. This condition simplifies to motion invertibility for the Haar transform as will be detailed in the next section.

2.2.1 Haar discrete wavelet transform

In this section, we consider the Haar DWT under motion compensation. The standard temporal Haar transform (non motion-compensated) equations are:

- Haar transversal analysis equations:

$$\begin{aligned} h_k[x] &= f_{2k+1}[x] - f_{2k}[x] \\ l_k[x] &= \frac{1}{2}f_{2k}[x] + \frac{1}{2}f_{2k+1}[x] \end{aligned}$$

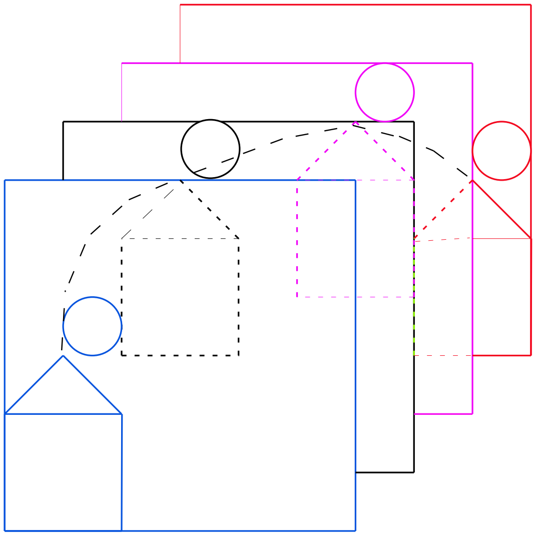


Figure 2.1: Filtering along motion trajectory

- Haar transversal synthesis equations:

$$\begin{aligned} f_{2k}[x] &= l_k[x] - \frac{1}{2}h_k[x] \\ f_{2k+1}[x] &= \frac{1}{2}h_k[x] + l_k[x] \end{aligned}$$

- Haar lifting-based analysis equations:

$$\begin{aligned} h_k[x] &= f_{2k+1}[x] - f_{2k}[x] \\ l_k[x] &= f_{2k}[x] + \frac{1}{2}h_k[x] \end{aligned}$$

- Haar lifting-based synthesis equations:

$$\begin{aligned} f_{2k}[x] &= l_k[x] - \frac{1}{2}h_k[x] \\ f_{2k+1}[x] &= h_k[x] + f_{2k}[x] \end{aligned}$$

Let $M_{k\to l}(x)$ denote the motion transformation from frame k to frame l. In an ideal case, $f_k[M_{k\to l}(x)] = f_l[x]$ holds. With this notation, the Haar motion-compensated transversal equations are:

- Haar motion-compensated transversal analysis equations:

$$\begin{aligned} h_k[x] &= f_{2k+1}[x] - \tilde{f}_{2k}[M_{2k\to 2k+1}(x)] \\ l_k[x] &= \frac{1}{2}f_{2k}[x] + \frac{1}{2}\tilde{f}_{2k+1}[M_{2k+1\to 2k}(x)] \end{aligned}$$

- Haar motion-compensated transversal synthesis equations:

$$\begin{aligned}\bar{f}_{2k}[x] &= l_k[x] - \frac{1}{2}\tilde{h}_k[M_{2k+1\to 2k}(x)]\\ \bar{f}_{2k+1}[x] &= \frac{1}{2}h_k[x] + \tilde{l}_k[M_{2k\to 2k+1}(x)]\end{aligned}$$

By substituting h_k and l_k into the synthesis equations, one can show that the conditions for perfect reconstruction are:

$$\begin{aligned}M_{2k\to 2k+1}(M_{2k+1\to 2k}(x)) &= x\\ M_{2k+1\to 2k}(M_{2k\to 2k+1}(x)) &= x\end{aligned}$$

These two equations mean that the motion transformation M needs to be invertible [1].

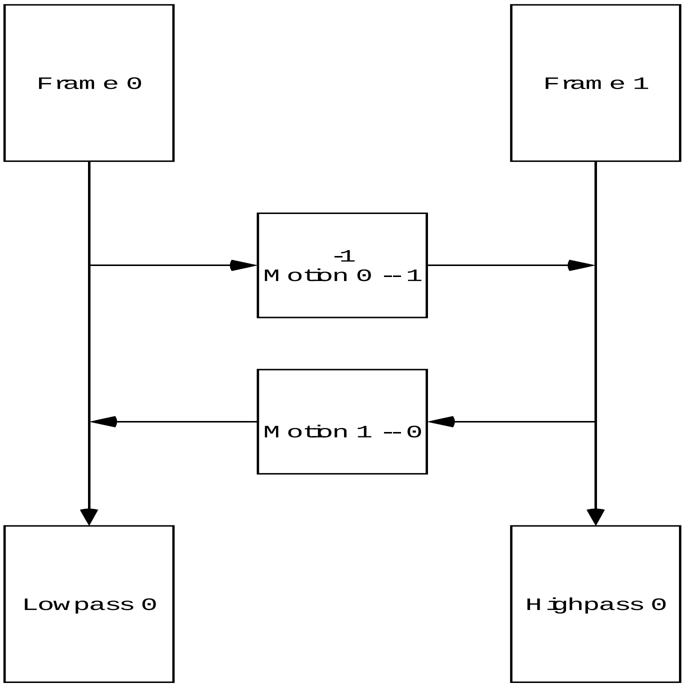


Figure 2.2: Use of motion in lifting-based Haar DWT

As for the Harr motion-compensated lifted transform, Fig. 2.2 show the role motion plays in the decomposition. The corresponding lifting equations are:

- Haar motion-compensated lifting-based analysis equations:

$$\begin{aligned}h_k[x] &= f_{2k+1}[x] - \tilde{f}_{2k}[M_{2k\to 2k+1}(x)]\\ l_k[x] &= f_{2k}[x] + \frac{1}{2}\tilde{h}_k[M_{2k+1\to 2k}(x)]\end{aligned}$$

- Haar motion-compensated lifting-based synthesis equations:

$$\begin{aligned}\bar{f}_{2k}[x] &= l_k[x] - \frac{1}{2}\tilde{h}_k[M_{2k+1\to 2k}(x)] \\ \bar{f}_{2k+1}[x] &= h_k[x] + \tilde{\bar{f}}_{2k}[M_{2k\to 2k+1}(x)]\end{aligned}$$

Note that in lifting implementations perfect reconstruction always holds, even if non-linearity is included into equations. However, the above Haar motion-compensated lifting equations produce exactly the same output as the Haar motion-compensated transversal equations only if motion model used is invertible [1].

2.2.2 Haar sub-optimal lifting

In the analysis equations of the motion-compensated lifted Haar transform, we first calculated the high subband (prediction step) and then we calculated the low subband based on the high subband. The motion fields used to calculate the high subband, i.e., $M_{2k\to 2k+1}$, were estimated in standard way by minimizing the motion-compensated prediction error:

$$\min_{M_{2k\to 2k+1}} \sum_x |f_{2k+1}[x] - \tilde{f}_{2k}[M_{2k\to 2k+1}(x)]|^2$$

Clearly, the resulting motion fields minimize energy in the high subband, a highly desirable effect. We call this optimal lifting.

However, we wondered what would happen had we not used the optimal lifting but instead, for example, lifting such that motion estimation criterion is not directly related to the energy of the high subband. In this case, motion fields ($M_{2k\to 2k+1}$) obtained by minimizing the criterion above are applied to calculating the low subband. Th high subband is computed by motion fields inverted from these motion fields and thus the high subband may contain a lot of energy.

- Haar motion-compensated sub-optimal lifting-based analysis equations:

$$\begin{aligned}l_k[x] &= \frac{1}{2}f_{2k}[x] + \frac{1}{2}\tilde{f}_{2k+1}[M_{2k+1\to 2k}(x)] \\ h_k[x] &= 2f_{2k+1}[x] - 2\tilde{l}_k[M_{2k\to 2k+1}(x)]\end{aligned}$$

- Haar motion-compensated sub-optimal lifting-based synthesis equations:

$$\begin{aligned}\bar{f}_{2k+1}[x] &= \frac{1}{2}h_k[x] + \tilde{l}_k[M_{2k\to 2k+1}(x)] \\ \bar{f}_{2k}[x] &= 2l_k[x] - \tilde{\bar{f}}_{2k+1}[M_{2k+1\to 2k}(x)]\end{aligned}$$

2.2.3 5/3 discrete wavelet transform

We also considered DWT based on the 5/3 filters as follows:

- 5/3 motion-compensated transversal analysis equations:

$$\begin{aligned} h_k[x] &= f_{2k+1}[x] - \frac{1}{2}(\tilde{f}_{2k}[M_{2k\to 2k+1}(x)] + \tilde{f}_{2k+2}[M_{2k+2\to 2k+1}(x)]) \\ l_k[x] &= \frac{3}{4}f_{2k}[x] + \frac{1}{4}(\tilde{f}_{2k-1}[M_{2k-1\to 2k}(x)] + \tilde{f}_{2k+1}[M_{2k+1\to 2k}(x)]) \\ & -\frac{1}{8}(\tilde{f}_{2k-2}[M_{2k-2\to 2k}(x)] + \tilde{f}_{2k+2}[M_{2k+2\to 2k}(x)]) \end{aligned}$$

- 5/3 motion-compensated transversal synthesis equations:

$$\begin{aligned} \bar{f}_{2k}[x] &= l_k[x] - \frac{1}{4}(\tilde{h}_{k-1}[M_{2k-1\to 2k}(x)] + \tilde{h}_k[M_{2k+1\to 2k}(x)]) \\ \bar{f}_{2k+1}[x] &= \frac{3}{4}h_k[x] - \frac{1}{8}(\tilde{h}_{k-1}[M_{2k-1\to 2k+1}(x)] + \tilde{h}_{k+1}[M_{2k+3\to 2k+1}(x)]) \\ & +\frac{1}{2}(\tilde{l}_k[M_{2k\to 2k+1}(x)] + \tilde{l}_{k+1}[M_{2k+2\to 2k+1}(x)]) \end{aligned}$$

For the 5/3 motion-compensated transversal DWT, sufficient conditions for perfect reconstruction are [1]:

$$\begin{aligned} M_{2k\to 2k-1}(M_{2k-1\to 2k}(x)) &= x \\ M_{2k\to 2k+1}(M_{2k+1\to 2k}(x)) &= x \\ M_{2k+1\to 2k}(M_{2k\to 2k+1}(x)) &= x \\ M_{2k+1\to 2k+2}(M_{2k+2\to 2k+1}(x)) &= x \end{aligned}$$

plus several conditions of the form: $M_{k\to n}(M_{n\to l}(x) = M_{k\to l}(x)$. The equations listed mean that the motion transform needs to be invertible, but the last condition is more general and requires that motion composition be well defined [1].

As for the 5/3 motion-compensated lifted DWT, we have:

- 5/3 motion-compensated lifting-based analysis equations:

$$\begin{aligned} h_k[x] &= f_{2k+1}[x] - \frac{1}{2}(\tilde{f}_{2k}[M_{2k\to 2k+1}(x)] + \tilde{f}_{2k+2}[M_{2k+2\to 2k+1}(x)]) \\ l_k[x] &= f_{2k}[x] + \frac{1}{4}(\tilde{h}_{k-1}[M_{2k-1\to 2k}(x)] + \tilde{h}_k[M_{2k+1\to 2k}(x)]) \end{aligned}$$

- 5/3 motion-compensated lifting-based synthesis equations:

$$\begin{aligned} \bar{f}_{2k}[x] &= l_k - \frac{1}{4}(\tilde{h}_{k-1}[M_{2k-1\to 2k}(x)] + \tilde{(h)}_k[M_{2k+1\to 2k}(x)]) \\ \bar{f}_{2k+1}[x] &= h_k[x] + \frac{1}{2}(\tilde{f}_{2k}[M_{2k\to 2k+1}[x] + \tilde{\bar{f}}_{2k+2}[M_{2k+2\to 2k+1}(x)]) \end{aligned}$$

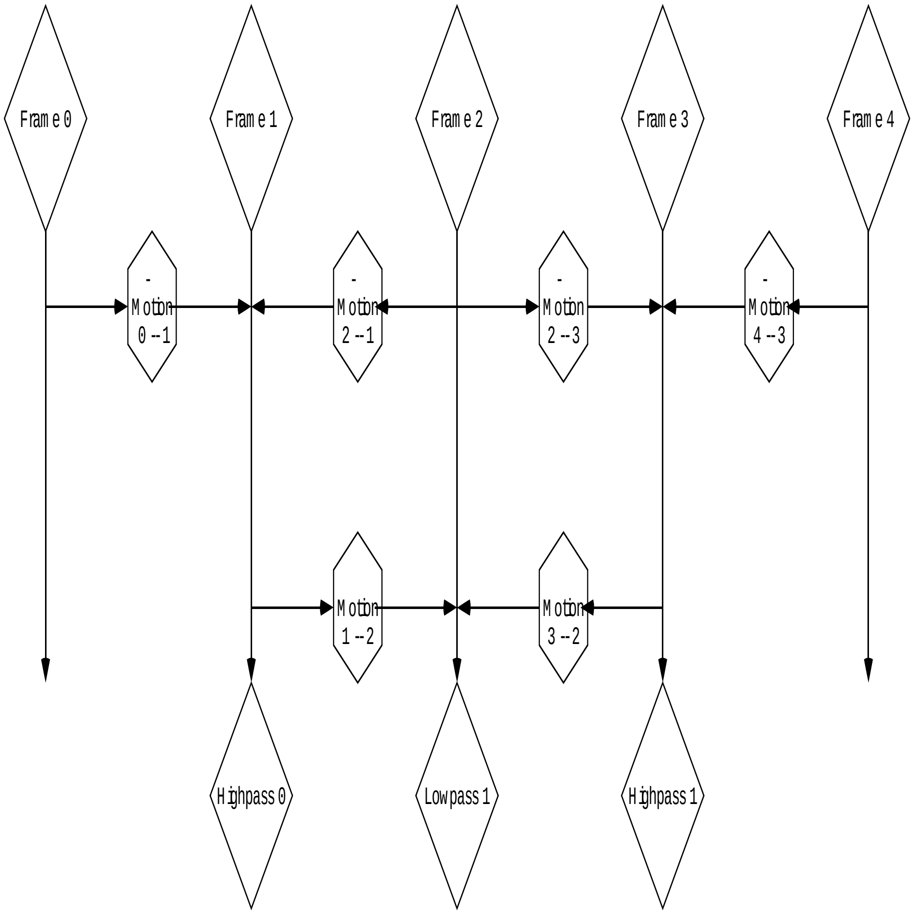


Figure 2.3: Use of motion in lifting-based 5/3 DWT.

In order that the 5/3 motion-compensated transversal and lifted DWTs be equivalent (outputs of their respective analysis and synthesis equations be the same), the conditions are again that motion be invertible and that motion composition be well defined [1].

Chapter 3

Motion models for temporal DWT

Clearly, in this project motion fields are an essential element of the overall compression scheme. Such motion fields are usually computed between two consecutive frames although they may also be derived from three or more frames. For a review of motion modeling and estimation paper by Stiller and Konrad is highly recommended [4].

The underlying model used in motion estimation from image sequences is intensity constancy along motion trajectories. Thus, for intensity ψ and displacement (motion) (d_x, d_y) we can write [6]:

$$\psi(x + d_x, y + d_y, t + d_t) = \psi(x, y, t)$$

After applying Taylor's expansion, we get:

$$\psi(x + d_x, y + d_y, t + d_t) = \psi(x, y, t) + \frac{\partial \psi}{\partial x} d_x + \frac{\partial \psi}{\partial y} d_y + \frac{\partial \psi}{\partial t} d_t.$$

Using the intensity constancy above, we obtain:

$$\frac{\partial \psi}{\partial x} d_x + \frac{\partial \psi}{\partial y} d_y + \frac{\partial \psi}{\partial t} d_t = 0$$

Finally, dividing the above equation by d_t we obtain:

$$\frac{\partial \psi}{\partial x} v_x + \frac{\partial \psi}{\partial y} v_y + \frac{\partial \psi}{\partial t} = 0$$

Since the above equation if scalar and we have two unknowns (velocities v_x and v_y), it is clear that we cannot determine both of them at the same time. We need to set up other constraint to make the equations complete. One way to impose constrains is to assume that neighboring pixels undergo the same motion. One example of using neighboring pixels is called block matching that we describe below.

3.1 Block matching

Block matching is a straightforward motion estimation method that is amenable to VLSI implementations. This method is widely adopted by current hybrid video compression standards. Block matching will be used here in motion-compensated DWTs as well.

However, motion fields resulting from block matching do not assure one to one mapping for pixel pairs between two adjacent frames; there will be multiple pixels from frame #1 pointing to the same position in frame #2 and some pixels in frame #2 will be never pointed to by any motion vectors. Clearly, the model is not invertible. Lifting-based approach and transversal approach will not result in the same output with this motion model.

Block matching is typically implemented as follows:

- frame #1 (called reference frame) is divided into non-overlapping blocks,
- a block in frame #1 is selected and a candidate vector is assigned to all pixels,
- an error metric is computed between intensities of the selected block in frame #1 and its translated version (by the selected motion candidate) in frame #2,
- vector with the lowest error for each block is selected and assigned to this block,
- the procedure is repeated for all blocks in frame #1.

Mean squared error is one of the most popular error metrics, which is also adopted in this work. Motion field is estimated by minimization of the sum of differences between intensities of block pairs :

$$E_m(\mathbf{d}_m) = \sum_{\mathbf{x}\in B_m} |\psi_2(\mathbf{x}+\mathbf{d}_m) - \psi_1(\mathbf{x})|^2$$

Example of vector field computed using this approach is shown in Fig. 3.3.

3.1.1 Unrestricted motion vectors

In order to provide a more reasonable description of motion in a given sequence and improve coding efficiency at image boundaries we allow motion vectors to point outside of the image [6]. In this project, images are expanded by one block in both directions by mirroring blocks at the boundary. Thus, motion vector candidates can point outside of the image by one block. Examples of the original and expanded image are shown in Fig. 3.1 and 3.2.

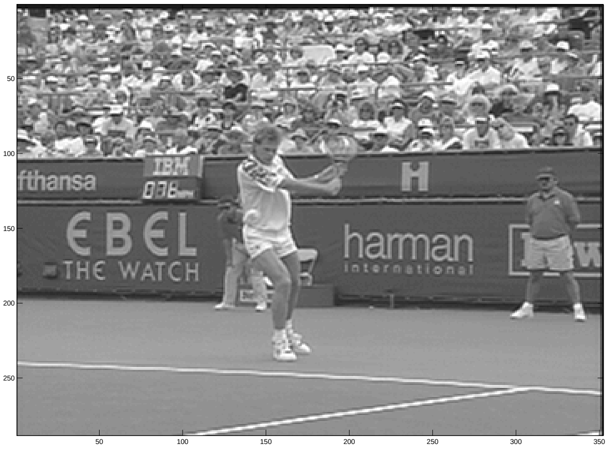


Figure 3.1: One frame from sequence "Stefan".

3.2 Sub-pixel search

In general, the motion between consecutive frames need not be composed of of integer horizontal and vertical components. For more precise motion fields, sub-pixel accuracy search is needed. Sub-pixel positioned points can be interpolated from the available intensities at grid points. In Fig. 3.4(a), points denoted by gray bullets are at sub-pixel positions, namely shifted by $(\frac{1}{4}, \frac{1}{4})$ from grid points (black bullets). Intensities at those sub-pixel positions may be needed in motion compensation. They can be interpolated from the known intensities at grid points (black bullets) using interpolation (linear, cubic, etc.). We use bilinear interpolation in this work.

3.3 Backward motion estimation

In order to implement analysis and then synthesis equations of motion-compensated temporal DWT, both forward and backward motion fields are needed. One approach is to estimate both motion fields independently (from frame #1 to frame #2, and from frame #2 to frame #1). The advantage is simplicity, but drawbacks are that two vector fields need to be transmitted and also that the two vector fields are unlikely to be inverses of each other.

An alternative is to compute the forward vector field directly but to estimate the backward field by some sort of inversion procedure. Since motion model used in block matching is not invertible, the inversion procedure cannot be exact but instead only an approximation. We developed a method based on nearest-neighbor interpolation of the forward vector field. The idea is as follows: each forward motion vector points to a position in frame #2 at which it can be reversed to point back to frame #1, and thus defining the backward vector field on an irregular set of positions. The main

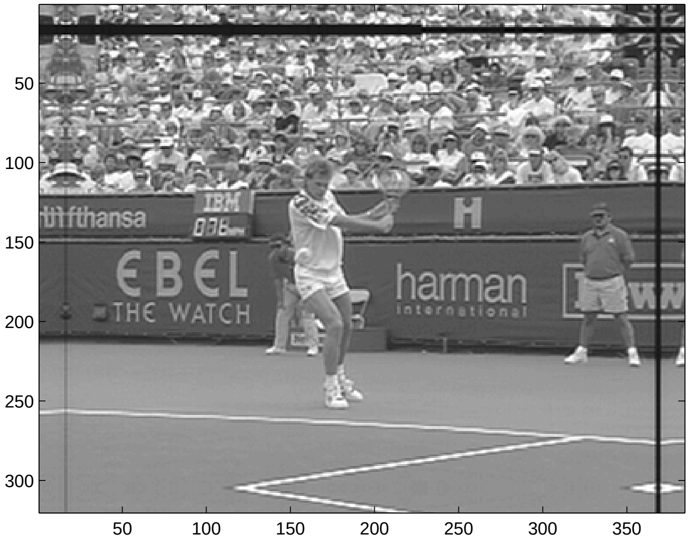


Figure 3.2: Frame from Fig. 3.1 after expansion.

issue is to perform irregular-to-regular grid interpolation [5]. While nearest-neighbor interpolation on pixel-precision vector fields is quite straightforward, it is a bit more complicated for sub-pixel accuracy.

Fig. 3.4(b) shows regular grid points (hashed) at which we need to estimate the backward motion vectors and irregular 1/4-pixel precision positions (black) that result from motion compensation; forward motion vectors are projected from frame #1 to frame #2 thus resulting in a set of irregular positions but always at 1/4-pixel precision (hashed). Based on the knowledge of x and y motion components at black points we need to recover these components at hashed points.

The nearest-neighbor interpolation algorithm that we developed can be described as follows (applied to both x and y components of motion vectors):

1. Initialize a all-zero matrix M in such a way that each element represents a full-pixel (grid) and sub-pixel position.

2. According to the forward motion field, assign the x component to those positions that are pointed to by the vector. To those positions that are not assigned a vector, assign a constant as a flag.

3. Depending on motion precision, initialize the vertical search range (HSV) and the horizontal search range (HSH). Both are adjustable.

4. Calculate how many points (`Num`) are within this search range and generate value pairs [-1,0],[0,-1],[1,0],...,[HSV,HSH] that denote search positions from the closest to the farthest away. Let `V` denote this ordered array of positions.

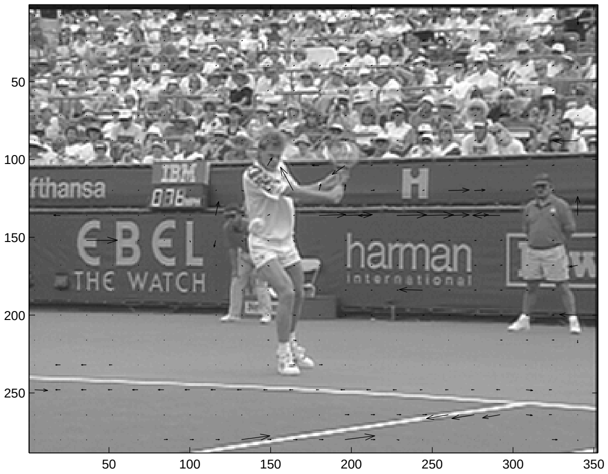


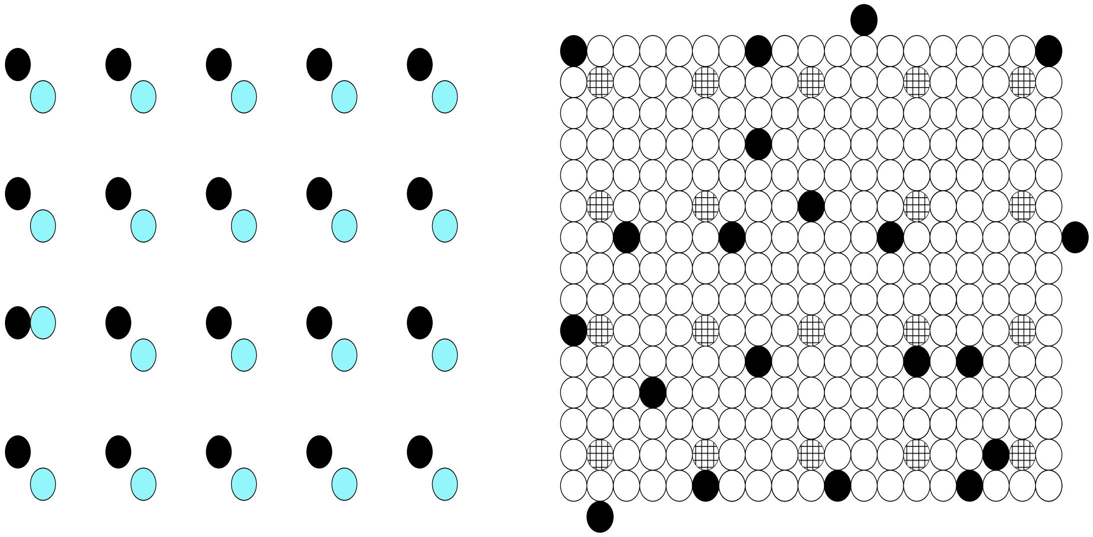
Figure 3.3: Motion field overlaid onto a frame of sequence “Stefan”.

5. On grid points, where we seek backward vectors, apply the following algorithm:

```
for i = grid points index
  for j = grid points index
      if M(i,j) == "flag"
          inx = 1;
          while inx<=Num AND M(i+V(inx,1),j+V(inx,2))=="flag"
          inx++
      end
      if inx<=Num
          M(i,j) = M(i+V(inx,1),j+V(inx,2));
      else
          M(i,j) = 'flag2'
      end
  end
end
```

6. Find those grid points that contain a ’flag2’, and perform the above algorithm with a very large search range.

Experiments show that a pair of motion fields obtained by nearest-neighbor interpolation described above are more invertible than two motion fields calculated directly. In other words, the absolute difference between two fields calculated directly is much larger. An example of independently estimated vector fields (horizontal component shown as intensity) is shown in Figs. 3.5-3.6. Note that the interpolated



(a) Sub-pixel search positions (b) Nearest-neighbor interpolation

Figure 3.4: (a) Sub-pixel search positions and (b) interpolation of motion vectors at regular positions (hashed) from vectors at irregular positions (black).

backward motion field is very similar to the estimated field although it is difficult to say which pair is closer to being invertible. In order to measure this objectively we developed the following invertibility error:

$$\epsilon_d = \sum_{\mathbf{x}} |\mathbf{d}^f(\mathbf{x}) - \tilde{\mathbf{d}}^b(\mathbf{x} + \mathbf{d}^f(\mathbf{x}))|$$

where $\mathbf{d}^f = [d_x^f, d_y^f]^T$ and $\mathbf{d}^b = [d_x^b, d_y^b]^T$ are forward and backward motion vectors, respectively, while $\tilde{\mathbf{d}}$ denotes interpolation, in our case bilinear, of x and y components of $\mathbf{d}$ at non-grid positions. Clearly, this error measures the sum of departures of points in frame #1 when each of them is projected onto frame #2 using the forward motion field and then back projected onto frame #1 using the backward motion field. A pair of motion fields being perfect inverses of each other would result in zero error ϵ_d.

In Table 3.1, we show the invertibility error ϵ_d for independently estimated and interpolated backward motion field for various sequences. Note the consistently lower error for the interpolated backward field as compared to the independently estimated field. As suggested by theoretical results from Chapter 2, we expect that a motion field pair with lower invertibility error should result in better compression performance.

3.4 Video coding performance measurement

To compare the impact of different motion fields on video compression performance, an objective criterion needs to be defined. To find an ideal criterion is very difficult. One popular way is to calculate PSNR, the peak-signal-to-noise ratio, in decibels

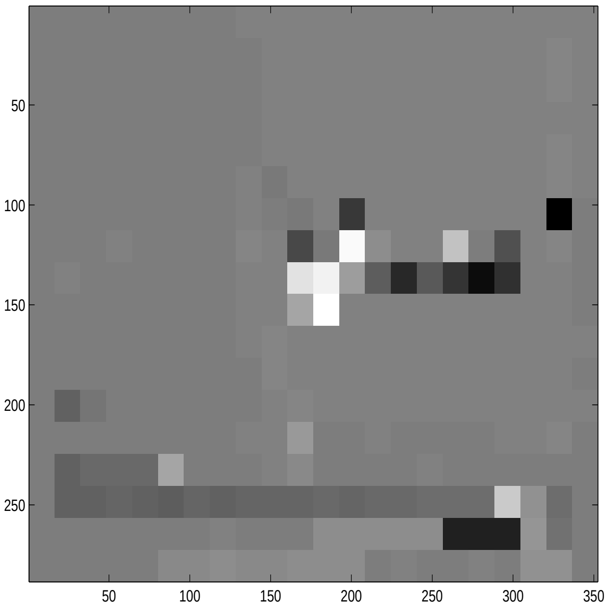


Figure 3.5: Horizontal component of forward motion field estimated from “Stefan”

(dB) [6] as follows:

$$PSNR = 10\log_{10}\frac{\psi_{max}^2}{\sigma_e^2},$$

where ψ_{max} is the maximum intensity in the data, and the variance σ_e^2 is computed as follows:

$$\sigma_e^2 = \frac{1}{N}\sum_k \sum_{m,n}(\psi_1(m,n,k) - \psi_2(m,n,k))^2.$$

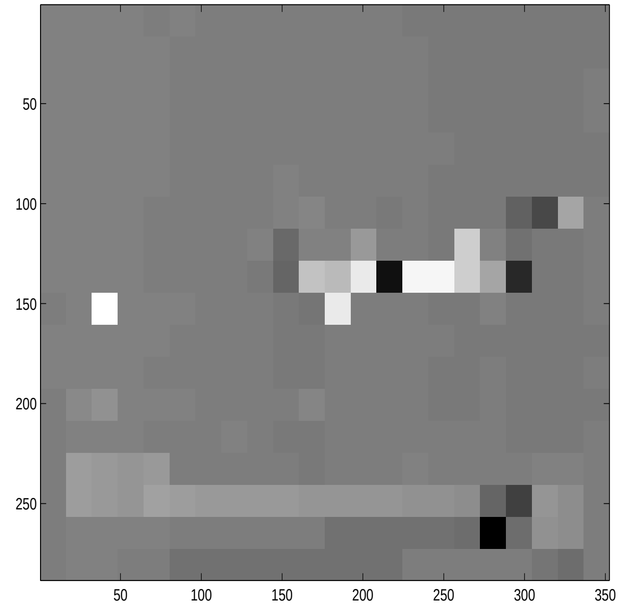


Figure 3.6: Horizontal component of backward motion field estimated from “Stefan”

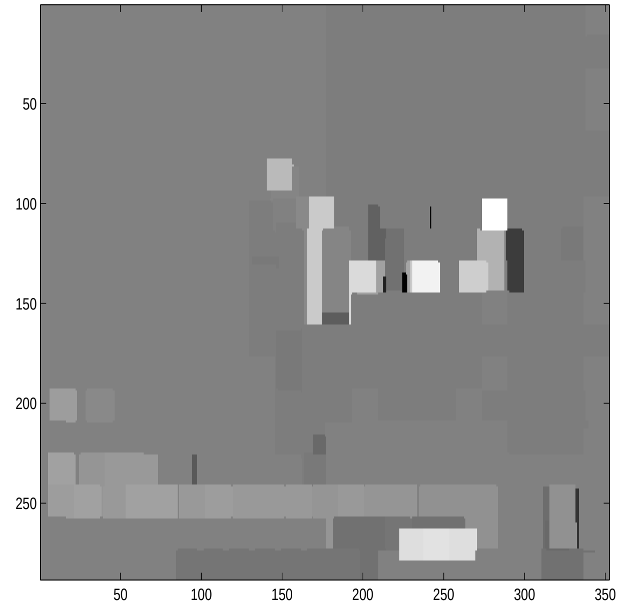


Figure 3.7: Horizontal component of backward motion field obtained by nearest-neighbor interpolation for “Stefan”

	1/4-pixel	1/2-pixel	full-pixel
“Stefan” (search range ±50)			
Two independent fields	2.3855/67.0981	2.4003/67.3284	2.3537/68.2659
Nearest-neighbor interp.	1.4295/45.0815	1.4182/45.0345	1.3793/45.7176
“Coastguard” (search range ±20)			
Two independent fields	0.0866/0.0298	0.0616/0.0365	0.0463/0.0475
Nearest-neighbor interp.	0.0066/0.0029	0.0038/0.0022	0.0044/0.0051
“Mobile” (search range ±25)			
Two independent fields	0.7827/12.6154	0.7913/12.6754	0.7739/12.7974
Nearest-neighbor interp.	0.3696/6.7768	0.3655/6.7487	0.3486/6.5995
“Akiyo” (search range ±20)			
Two independent fields	0.0310/0.1541	0.0288/0.1591	0.0153/0.1482
Nearest-neighbor interp.	0.0113/0.1241	0.0104/0.1197	0.0093/0.1239
“Hall” (search range ±25)			
Two independent fields	1.1953/11.5678	1.2023/11.5734	1.1378/11.9603
Nearest-neighbor interp.	0.4529/5.1760	0.4528/5.1810	0.4312/5.4574

Table 3.1: Comparison of invertibility error ϵ_d for various sequences; two independent motion fields estimated using block matching, or forward motion field estimated using block matching and backward recovered using nearest-neighbor interpolation.

Chapter 4

Experiments and results

4.1 Experiment setup

In this chapter, experimental procedure and results are provided, and conclusions are drawn.

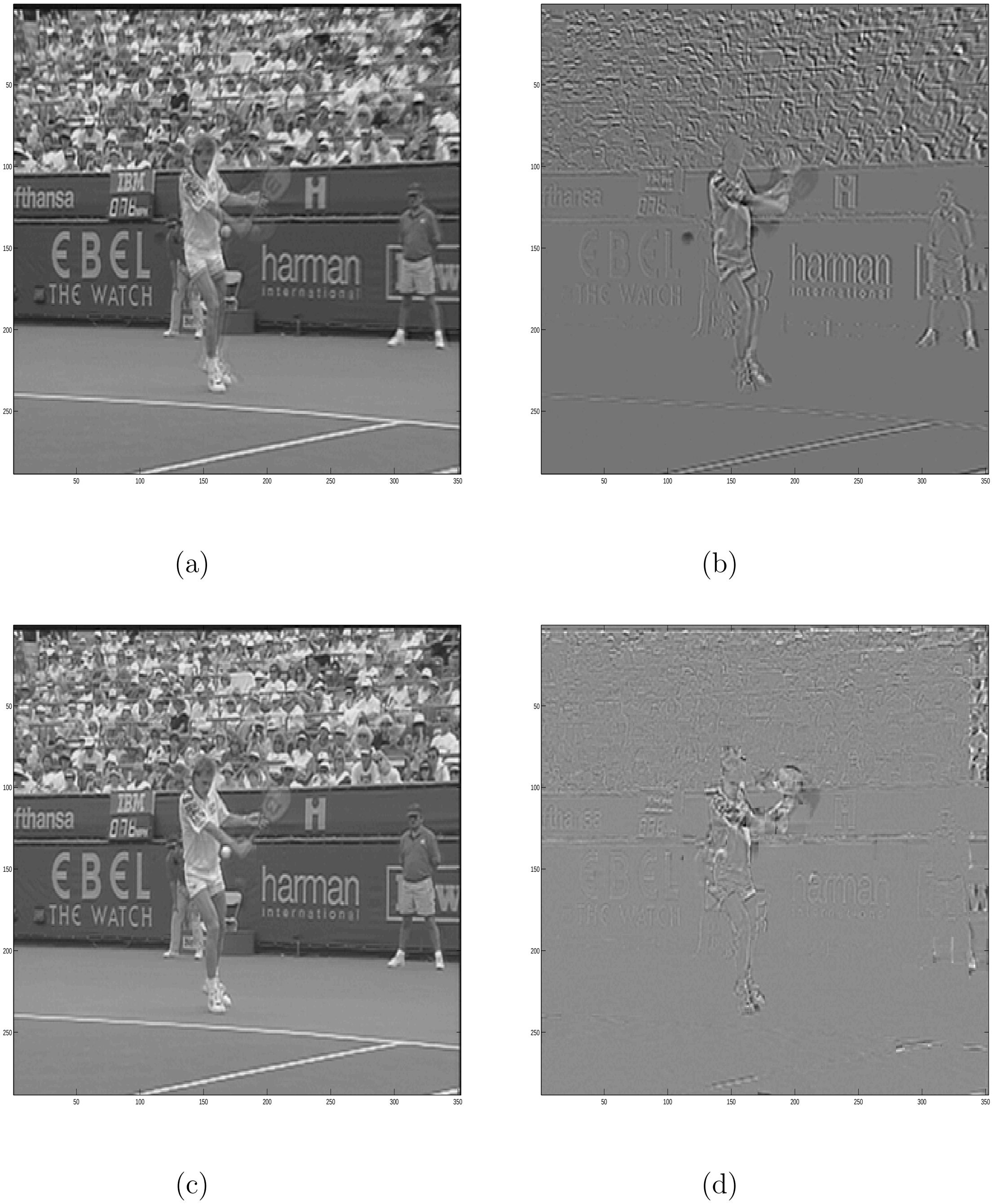
We conducted experiments are on CIF-resolution video sequences at 30 frames per second. Each frame contains 288×352 pixels. After the temporal transform, subband frames are spatially compressed by JPEG-2000 still-image coder. Then, video sequences are reconstructed by temporal synthesis filters. Single-level motion-compensated temporal Haar transform was performed by three different approaches: transversal approach, lifting-based approach and sub-optimal lifting-based approach. Backward motion fields were obtained by two different methods: direct estimation of forward and backward vector fields, and by inversion of directly-estimated forward motion field by means of nearest-neighbor interpolation.

Video sequences were coded at the bit rate of 500kbps. 90% of all bits were assigned to the low subband and 10% of all bits were assigned to the high subband. After decoding by JPEG-2000 decoder, corresponding motion-compensated synthesis filters were used to reconstruct each video sequence. All of the results are obtained by neglecting the cost of coding motion fields. Differences result only from motion fields and temporal filters.

As mentioned before, block matching is used to estimate motion. For all sequences we used $(-25, +25)$-pixel search range. Full-, half- and quarter-pixel accuracy motion vectors have been computed using bilinear interpolation. The block size was 16×16. All intensity interpolations in analysis and synthesis equations were carried out using linear interpolation.

4.2 Experimental results

For the frame from Fig. 3.1 we show the low and high subbands with no motion compensation (Fig. 4.1(a-b)), and with motion compensation (Fig. 4.1(c-d)). Note the reduced energy in the high subband when motion compensation is used.



(a)

(b)

(c)

(d)

Figure 4.1: Lowpass (a,c) and highpass (b,d) subbands for frame of "Stefan" from Fig. 3.1 after one level of 5/3 temporal decomposition without motion compensation (a,b) and with motion compensation (c,d).

4.3 Motion compensation benefits

The first experiment is to evaluate how much motion compensation will benefit the compression performance, and what is the improvement due to the use of 5/3 filters compared to simple Haar filters. Table 4.1 shows that the compression performance can be improved by introducing motion compensation rather than no motion compensation. For Haar filters the improvement introduced by motion compensation is 1.79dB. The improvement due to 5/3 filters instead of Haar filters is an additional 0.3dB. Results are obtained by transversal filters and nearest-neighbor interpolation at quarter-pixel precision. Experiment was performed on 30 frames of sequence ”Stefan” starting at the third frame.

Stefan	PSNR	Energy in high subband
Haar no motion	22.5778	1.3584e9
Haar with motion	+1.7945	1.5460e8
5/3 with motion	+2.0950	1.2082e8

Table 4.1: Coding performance expressed as PSNR for Haar DWT without motion, with motion and 5/3 with motion on “Stefan”.

4.4 Optimal and sub-optimal lifting

In this experiment we evaluate the difference between optimal and sub-optimal lifting (Table 4.2). Numbers after PSNR values are total energies in high subbands (prediction error). Results are obtained for the Haar filters and show that optimal filters outperform sub-optimal filters by 0.3dB. The experiment was performed on 30 frames of sequence ”Stefan” starting from the third frame and 30 frames of sequence ”Foreman” starting from the 203rd frame.

4.5 Lifting versus transversal implementation for independent two motion fields and nearest-neighbor inversion

The main object of this project was to examine how motion invertibility affects video compression performance. As mentioned before, motion field pair with backward field obtained through nearest-neighbor interpolation is, by definition (and confirmed by Table 3.1), closer to invertibility than motion field pair with backward field estimated directly. Since both the lifting approach is closer to transversal approach, and the transversal approach is closer to perfect reconstruction when motion is more invertible, a better performance can be expected of nearest-neighbor interpolation over two

Stefan	1/4 precision	1/2 precision	full precision
Two fields optimal	24.2810(1.5460e8)	24.1936(1.7100e8)	24.0151(2.2347e8)
Two fields sub-optimal	23.9454(5.0645e8)	23.8871(5.2697e8)	23.7690(5.4084e8)
Nearest-neighbor optimal	24.3156(1.5460e8)	24.2845(1.7100e8)	24.1619(2.2347e8)
Nearest-neighbor sub-optimal	24.0145(6.0809e8)	24.0104(6.1011e8)	23.9599(5.9331e8)

Foreman	1/4 precision	1/2 precision	full precision
Two fields optimal	30.5808(4.8534e7)	30.5020(5.2336e7)	30.4111(6.5041e7)
Two fields sub-optimal	30.1866(1.7329e8)	30.1666(1.7859e8)	30.0419(1.7627e8)
Nearest-neighbor optimal	30.6918(4.8534e7)	30.6634(5.2336e7)	30.5881(6.5041e7)
Nearest-neighbor sub-optimal	30.1895(3.0861e8)	30.1467(3.1682e8)	30.0921(3.1656e8)

Table 4.2: Coding performance expressed as PSNR for Haar optimal and sub-optimal lifting filters on “Stefan” and “Foreman”. Energy in the high subband is shown in parentheses.

fields estimated independently. Results in Table 4.3 show that motion field pairs with backward fields inverted by nearest-neighbor interpolation outperform motion field pairs with backward fields estimated directly by up to 0.15dB. In Table 4.4, we also show results for “Stefan” using the 5/3 filters. As can be seen, the 5/3 filters outperform the Haar filters by about 0.3dB in both transversal and lifted implementations with two independent motion fields and nearest-neighbor interpolation.

4.6 Conclusions and acknowledgments

4.6.1 Conclusions and future directions

Invertibility of motion models plays an important role in wavelet temporal transforms. In this work, simple interpolation method to construct backward motion field from a forward one has been proposed. This has lead to motion fields that are closer to forming an invertible pair than a pair of motion fields estimated directly. Experiments have been carried out to evaluate compression performance of both approaches to the construction of motion field pairs under both transversal and lifting-based temporal DWTs. Results show that motion field pairs that are more invertible perform better than motion field pairs that are less invertible. This confirms the theoretical results recently derived.

Future work includes studying this issue for longer filters like 9/7 and also multistage decomposition. Motion fields can be estimated by mesh model and inverted by more advanced spline based method. These are all very interesting experiments to work on. In block matching, different block size and motion precision are also worth testing. In experiment experiments reported here, we used a simple JPEG-2000 still image coder and we coded each subband frame separately. Bits assigned to each

Stefan	1/4-pixel	1/2-pixel	full-pixel
Two fields transversal	24.2758(1.5460e8)	24.2203(1.7100e8)	24.1026(2.2347e8)
Nearest-neighbor transversal	24.3723(1.5460e8)	24.3115(1.7100e8)	24.2231(2.2347e8)
Improvement	0.0965	0.0912	0.1205
Two fields lifting	24.2810(1.5460e8)	24.1936(1.7100e8)	24.0151(2.2347e8)
Nearest-neighbor lifting	24.3156(1.5460e8)	24.2845(1.7100e8)	24.1619(2.2347e8)
Improvement	0.0346	0.0909	0.1468

Coastguard	1/4-pixel	1/2-pixel	full-pixel
Two fields transversal	28.2011(5.1550e7)	28.1954(5.5213e7)	28.1157(6.5637e7)
Nearest-neighbor transversal	28.2243(5.1550e7)	28.1980(5.5213e7)	28.1502(6.5637e7)
Improvement	0.0232	0.0026	0.0345
Two fields lifting	28.1646(5.1550e7)	28.1540(5.5213e7)	28.1496(6.5637e7)
Nearest-neighbor lifting	28.1884(5.1550e7)	28.1619(5.5213e7)	28.1543(6.5637e7)
Improvement	0.0238	0.0079	0.0047

Mobile	1/4-pixel	1/2-pixel	full-pixel
Two fields transversal	20.8407(1.7329e8)	20.7865(2.1365e8)	20.6967(3.6593e8)
Nearest-neighbor transversal	20.8593(1.7329e8)	20.8174(2.1365e8)	20.7167(3.6593e8)
Improvement	0.0186	0.0309	0.0200
Two fields lifting	20.8166(1.7329e8)	20.7809(2.1365e8)	20.6995(3.6593e8)
Nearest-neighbor lifting	20.8327(1.7329e8)	20.7903(2.1365e8)	20.7421(3.6593e8)
Improvement	0.0161	0.0094	0.0426

Akiyo	1/4-pixel	1/2-pixel	full-pixel
Two fields transversal	37.1277(3.0422e6)	37.0983(3.7547e6)	37.0566(4.7553e6)
Nearest-neighbor transversal	37.1531(3.0422e6)	37.1067(3.7547e6)	37.0551(4.7553e6)
Improvement	0.0254	0.0084	-0.0015
Two fields lifting	37.1617(3.0422e6)	37.1029(3.7547e6)	37.0508(4.7553e6)
Nearest-neighbor lifting	37.1820(3.0422e6)	37.1189(3.7547e6)	37.0480(4.7553e6)
Improvement	0.0203	0.0160	-0.0028

Hall	1/4-pixel	1/2-pixel	full-pixel
Two fields transversal	31.5369(3.9562e7)	31.5337(4.1639e7)	31.4347(4.5563e7)
Nearest-neighbor transversal	31.6155(3.9562e7)	31.6071(4.1639e7)	31.5382(4.5563e7)
Improvement	0.0786	0.0734	0.1035
Two fields lifting	31.6221(3.9562e7)	31.6226(4.1639e7)	31.5369(4.5563e7)
Nearest-neighbor lifting	31.6950(3.9562e7)	31.6991(4.1639e7)	31.6345(4.5563e7)
Improvement	0.0729	0.0765	0.0976

Table 4.3: Coding performance expressed as PSNR for Haar lifting versus transversal under two motion fields and nearest-neighbor inversion for “Stefan”, “Coastguard”, “Mobile”, “Akiyo”, “Hall”. Energy in the high subband is shown in parentheses.

Stefan	1/4-pixel	1/2-pixel	full-pixel
Two fields transversal	24.5141(1.0994e8)	24.5345(1.1658e8)	24.3916(129721701)
Nearest-neighbor transversal	24.6728(1.2082e8)	24.6544(1.2844e8)	24.4970(1.4547e8)
Improvement	0.1587	0.1199	0.1054
Two fields lifting	24.5326(1.0994e8)	24.5112(1.1658e8)	24.3600(129721701)
Nearest-neighbor lifting	24.6779(1.2082e8)	24.6563(1.2844e8)	24.4391(1.4547e8)
Improvement	0.1453	0.1451	0.0791

Table 4.4: Coding performance expressed as PSNR for 5/3 lifting versus transversal under two motion fields and nearest-neighbor inversion for 30 frames of ”Stefan”. Energy in the high subband is shown in parentheses.

frame were fixed. A video coder that uses more advanced bit assignment algorithm would prove more beneficial.

4.6.2 Acknowledgments

I would like to express my sincerest gratitude to my adviser, Prof. Janusz Konrad and to my fellow student Nikola Božinović. Without their careful advising and guidance, I couldn’t have been so close to the cutting edge of video processing technology and learn so much during the past semester.

Bibliography

[1] J. Konrad, "Transversal versus lifting approach to motion-compensated temporal discrete wavelet transform of image sequences: equivalence and tradeoffs," in *Proc. SPIE Visual Communications and Image Process.*, vol. 5308, Jan. 2004.

[2] A. Secker, *Motion-adaptive transforms for highly scalable video compression.* PhD thesis, University of New South Wales, School of Electr. Eng. and Telecom., Aug. 2004.

[3] A. Secker and D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression," *IEEE Trans. Image Process.*, vol. 12, pp. 1530–1542, Dec. 2003.

[4] C. Stiller and J. Konrad, "Estimating motion in image sequences: A tutorial on modeling and computation of 2D motion," *IEEE Signal Process. Mag.*, vol. 16, pp. 70–91, July 1999.

[5] C. Vázquez, E. Dubois, and J. Konrad, "Reconstruction of irregularly-sampled images in spline spaces," *IEEE Trans. Image Process.*, Oct. 2003 (submitted).

[6] J. O. Y. Wang and Y.-Q. Zhang, *Video Processing and Communciations.* Prentice Hall, 2002.