Highly Scalable Video Compression with Scalable Motion Coding

Andrew Secker and David Taubman *

Abstract

A scalable video coder cannot be equally efficient over a wide range of bit-rates unless both the video data and the motion information are scalable. We propose a wavelet-based, highly scalable video compression scheme with rate-scalable motion coding. The proposed method involves the construction of quality layers for the coded sample data and a separate set of quality layers for the coded motion parameters. When the motion layers are truncated, the decoder receives a quantized version of the motion parameters used to code the sample data. The effect of motion parameter quantization on the reconstructed video distortion is described by a linear model. The optimal tradeoff between the motion and subband bit-rates is determined after compression. We propose two methods to determine the optimal tradeoff, one of which explicitly utilizes the linear model. This method performs comparably to a brute force search method, reinforcing the validity of the linear model itself. Experimental results indicate that the cost of scalability is small. In addition, considerable performance improvements are observed at low bit-rates, relative to lossless coding of the motion information.

1 Introduction

Highly scalable video coders based on three-dimensional (3D) discrete wavelet transforms (DWT) offer high coding efficiency within a natural multiresolution representation. For maximum com-
pression efficiency and visual quality it is essential that the temporal transform adapts effectively to exploit motion. This requires realistic motion modelling and sufficiently precise motion parameters.

Motion parameters are usually coded losslessly as side-information. However, the tradeoff between the volume of motion information and the efficiency of motion compensation has been widely recognized. In non-scalable coders, various techniques have been used to optimize the number of bits spent on motion parameters for a target bit-rate [1, 2, 3]. In scalable video coding, similar techniques have been employed to improve the efficiency of motion compensation [4], but the motion overhead is invariably regarded as a non-scalable element of the compression system.

Without some form of scalability in the motion information, scalable coders cannot operate equally efficiently over a wide range of bit-rates. In particular, at low bit-rates the motion information can consume an undue proportion of the available bits. Furthermore, at low bit-rates the video is often decoded at a reduced spatial resolution, in which case high-precision motion information is of little benefit. Conversely, at high bit-rates the motion cost becomes insignificant, so that higher compression may be possible with a more precise representation of the motion.

In this paper we propose a scalable video coding scheme with rate-scalable motion information. We extend and elaborate on preliminary work previously published in [5]. Motion parameters required by a motion-adaptive temporal transform are scalably coded using techniques similar to those embodied in the JPEG2000 image compression standard [6, 7]. In the proposed scheme, video frames are compressed using unquantized motion parameters, but the decoder receives and uses only the most appropriate motion parameter quality layer.

Quantization of the motion parameters introduces distortion in the video frames. For temporal transforms with finite support, this distortion is bounded. By contrast, however, post-compression quantization of the motion parameters used by a predictive feedback scheme results in distortion with infinite temporal extent, an effect similar to prediction drift.

Motion parameters have traditionally been coded losslessly due to the non-linear interaction between motion and sample data. However, for all interesting combinations of motion bit-rate and
sample bit-rate, the relationship between motion error and the resulting video distortion turns out to be close to linear. An additive distortion model, combined with a layered motion bit-stream, allows an optimal balance between motion and subband bit-rates to be determined after compression.

The temporal transform used in this work is described in Section 2. In Section 3 we derive a linear model for the distortion introduced by motion error during temporal synthesis. In Section 4, we use this model to show that a rate-distortion optimized layered motion representation can be constructed independently of the scalable video sample representation. The optimal balance between motion and sample data bit-rates may then be found after compression, according to the particular bit-rate and spatial resolution at which the compressed video is decoded.

Our proposed scalable motion parameter coding scheme is described in Section 5. We show that the application of wavelet transformations to the motion vector fields, prior to embedded block coding and optimized layering, improves video coding efficiency.

In Section 6 we present two practical methods for determining the optimal rate-allocation between motion and subband data. We first describe a search-based method which involves several reconstructions using different motion quality layers. A second, less computationally intensive method, directly determines the sensitivity of the video sequence to motion error, allocating motion bits accordingly. Experimental results in Section 7 show that the cost of introducing motion scalability is actually very small. In addition, we observe significant improvements in low bit-rate compression performance relative to lossless motion coding.

2 LIMAT Framework and Motion

The LIMAT (Lifting-based Invertible Motion Adaptive Transform) framework enables the construction of efficient motion-adaptive temporal transforms, with any motion model and any wavelet kernel. This framework is based on the lifting representation for the DWT, with motion compensation operations applied within the lifting steps to exploit motion.

Earlier research indicates that increased compression is possible using the 5/3 wavelet kernel,
in comparison to the Haar [8]. However, the longer filters often result in higher motion overhead, reinforcing the need for scalable motion data. Prior research has also shown that continuously deformable mesh motion models yield higher coding efficiency in comparison to typical block-based models. For these reasons, this work focuses on the 5/3 transform with triangular mesh motion modelling. Note, however, that the results and methods described in this paper are equally applicable to other motion models and wavelet kernels.

The lifting steps for the original 5/3 wavelet transform are

\[
\begin{align*}
    h_k[n] &= x_{2k+1}[n] - \frac{1}{2}(x_{2k}[n] + x_{2k+2}[n]) \\
    l_k[n] &= x_{2k}[n] + \frac{1}{4}(h_{k-1}[n] + h_k[n])
\end{align*}
\]

where \( x_k[n] \equiv x_k[n_1, n_2] \) denotes the samples of frame \( k \) from the original video sequence. Similarly, \( h_k[n] \equiv h_k[n_1, n_2] \) and \( l_k[n] \equiv l_k[n_1, n_2] \) denote the high-pass and low-pass subband frames, respectively. The first step is generally known as the ‘prediction’ step and the second lifting step is known as the ‘update’ step.

As mentioned, we modify the lifting steps to exploit motion. Let \( W_{k_1 \rightarrow k_2} \) denote a motion-compensated mapping of frame \( k_1 \) onto the coordinate system of frame \( k_2 \), so that \( W_{k_1 \rightarrow k_2}(x_{k_1})[n] \approx x_{k_2}[n] \), for all \( n \). The lifting steps become

\[
\begin{align*}
    h_k &= x_{2k+1} - \frac{1}{2}(W_{2k \rightarrow 2k+1}(x_{2k}) + W_{2k+2 \rightarrow 2k+1}(x_{2k+2})) \\
    l_k &= x_{2k} + \frac{1}{4}(W_{2k-1 \rightarrow 2k}(h_{k-1}) + W_{2k+1 \rightarrow 2k}(h_k))
\end{align*}
\]

The high-pass frames are essentially the residual from bi-directional motion-compensated prediction, based on the even-indexed original frames.

The synthesis system simply applies the same motion-compensated lifting steps in reverse order, with the sign negated.

\[
\begin{align*}
    x_{2k} &= l_k - \frac{1}{4}(W_{2k-1 \rightarrow 2k}(h_{k-1}) + W_{2k+1 \rightarrow 2k}(h_k)) \\
    x_{2k+1} &= h_k + \frac{1}{2}(W_{2k \rightarrow 2k+1}(x_{2k}) + W_{2k+2 \rightarrow 2k+1}(x_{2k+2}))
\end{align*}
\]
Evidently, the synthesis system is able to recover the original video sequence from its subbands, so long as the motion mappings $W_{k_1 \rightarrow k_2}$ used during analysis are identical to those used during synthesis.

In each stage of the 5/3 transform, both forward and backward motion mappings are required between every pair of consecutive frames. One is used in a prediction step, and the other is used in an update step. As described in [8], it is sufficient to code and estimate only those mappings required for the prediction steps, since the others may be recovered by inversion. So long as mesh folding is avoided, $W_{2k \rightarrow 2k \pm 1}$ provides a 1-1 mapping between locations in frames $2k$ and $2k \pm 1$, which may be inverted to obtain the corresponding update mapping operator, $W_{2k \pm 1 \rightarrow 2k}$. The inverse is not always well-defined at image boundaries, where we employ an extension technique which ensures that global affine motion is preserved. Such details, however, are of little direct relevance to the themes of this paper.

3 Implications of Motion Field Quantization

In this section we quantify the effects of motion error on reconstructed video quality. After considering the case of a single motion compensation operator, with constant motion error, we extend our discussion to include a complete motion-compensated temporal synthesis system with arbitrary motion error.

The relationship between motion vector mean squared error (MSE) and the resulting video distortion depends primarily on the power spectral properties of the video data. In fact, this relationship turns out to be approximately linear for all optimal combinations of motion and sample bit-rates. By “optimal” we mean those combinations of motion and sample bit-rates which are found to minimize the reconstructed video distortion at each selected total bit-rate. These observations motivate the scalable motion coding scheme proposed in Section 5. In particular, they justify the construction of a layered motion parameter representation that is independent of the layered subband sample representation. The most efficient tradeoff between motion and sample information
can be determined after coding, according to the power spectral properties of the reconstructed video frames. The reconstructed video power spectra mainly depend on the bit-rate and spatial resolution at which the video data is accessed, suggesting that efficient rate-allocation need only account for these reconstruction parameters. This is demonstrated by the rate-allocation schemes presented in Section 6.

### 3.1 Motion Compensation with Constant Motion Error

Consider the motion warping operation, \( y[n] = W(x)[n] \), and suppose that \( W \) is affected by quantization of its motion parameters, resulting in the quantized operator \( W' \), which produces \( y'[n] = W'(x)[n] \). Let us first consider the effect of a constant displacement error, \( \delta \). Assuming ideal motion compensated interpolation, \( \delta \) contributes a linear phase shift to the Fourier transform, \( \hat{y}(\omega) \), of \( y[n] \). Specifically, \( \hat{y}'(\omega) = \hat{y}(\omega)e^{-j\omega^t\delta} \). Here, \( \omega \) denotes two dimensional frequency (in radians) and \( \omega^t\delta = \omega_1\delta_1 + \omega_2\delta_2 \) is the inner product between \( \delta \) and \( \omega \). The total squared error \( D_y \), between \( y[n] \) and \( y'[n] \), is given by

\[
D_y = \frac{1}{(2\pi)^2} \int \int S_y(\omega) \cdot |1 - e^{-j\omega^t\delta}|^2 d\omega_1 d\omega_2
\]

where \( S_y(\omega) \) is the energy density of \( y \). The Taylor series expansion of \( |1 - e^{-j\omega^t\delta}|^2 \) yields the following polynomial in \((\omega^t\delta)^2\)

\[
|1 - e^{-j\omega^t\delta}|^2 = 2 - 2\cos(\omega^t\delta) = \frac{2(\omega^t\delta)^2}{2!} - \frac{2(\omega^t\delta)^4}{4!} + \frac{6(\omega^t\delta)^6}{6!} - ... \tag{3}
\]

The higher order terms in equation (3) are insignificant for small values of \( \omega^t\delta \), allowing us to make the following approximation, for small \( \delta \).

\[
D_y \approx \frac{1}{(2\pi)^2} \int \int S_x(\omega) \cdot (\omega^t\delta)^2 d\omega \tag{4}
\]

In equation (4) we have made use of the fact that \( S_y(\omega) \approx S_x(\omega) \). This is true so long as the motion warping operator \( W \) is not excessively expansive or contractive, and is implemented using sufficiently high order interpolators. Equation (4) may be expressed as

\[
D_y \approx \delta_1^2\Psi_{1,x} + \delta_2^2\Psi_{2,x} + \delta_1\delta_2\Psi_{3,x} \tag{5}
\]
where $\Psi_{1,x}, \Psi_{2,x}$ and $\Psi_{3,x}$ represent three motion sensitivity factors, defined by

$$
\Psi_{1,x} = \frac{1}{(2\pi)^2} \int \int S_x(\omega) \omega_1^2 d\omega \quad (6)
$$

$$
\Psi_{2,x} = \frac{1}{(2\pi)^2} \int \int S_x(\omega) \omega_2^2 d\omega \quad (7)
$$

$$
\Psi_{3,x} = \frac{2}{(2\pi)^2} \int \int S_x(\omega) \omega_1 \omega_2 d\omega \quad (8)
$$

The $\omega_1^2, \omega_2^2$ and $\omega_1 \omega_2$ terms in the respective motion sensitivity expressions for $\Psi_{1,x}, \Psi_{2,x}$ and $\Psi_{3,x}$, indicate a strong dependence on the frame’s high frequency energy content. This is not surprising, since spatial edges are affected by motion errors much more than smooth spatial regions. The motion sensitivity also depends significantly on the range of spatial frequencies being integrated in equations (6), (7) and (8). For example, since power spectra for natural images typically decay roughly as $\|\omega\|^{-2}$ [9], if the integrals are from $-R \leq \omega_1, \omega_2 \leq R$ then $\Psi_{1,x}, \Psi_{2,x}$ and $\Psi_{3,x}$ are each proportional to $R^2$. This important observation means that we can expect the video frame to be particularly resilient to motion error when reconstructed at reduced spatial resolutions.

Quantization can also affect the frame’s power spectrum, since efficient lossy compression often involves quantizing high spatial frequencies more heavily than low spatial frequencies. In fact, at low bit-rates, high frequency components are often completely eliminated so that the motion sensitivity of the quantized frame is significantly less than the original.

Expressing $\delta$ in polar coordinates, equation (5) may be written as

$$
D_y \approx \|\delta\|^2 \Psi_x(\theta_\delta)
$$

where $\Psi_x(\theta_\delta)$, given below, is the motion sensitivity, expressed as a function of orientation $\theta_\delta$ of the motion error $\delta$.

$$
\Psi_x(\theta_\delta) = (\Psi_{1,x} \cos^2(\theta_\delta) + \Psi_{2,x} \sin^2(\theta_\delta) + \Psi_{3,x} \cos(\theta_\delta) \sin(\theta_\delta))
$$

Natural images often exhibit roughly isotropic power spectra. This means the sensitivity to motion error can be adequately represented by the average sensitivity $\Psi_x$ over all motion error
orientations.

\[
\Psi_x = \frac{1}{2\pi} \int_{-\pi}^{\pi} \Psi_x(\theta_\delta) d\theta_\delta = \frac{\Psi_{1,x} + \Psi_{2,x}}{2}
\]

(9)

This results in the following linear distortion model, for small and constant motion errors.

\[
D_y \approx \|\delta\|^2 \Psi_x
\]

(10)

We could formulate a more accurate model by including an explicit representation for the motion sensitivity in each spatial direction. While this may reduce model error for sequences with highly anisotropic power spectra, we choose not to pursue the possibility further in the present paper.

If we neglect the higher order terms in the polynomial (3), the model will tend to over-estimate frame distortion for large values of \(\|\delta\|^2\). However, for rate-allocation purposes, an accurate model is only required for motion errors on the order of that which represents the optimal tradeoff. Of course, when the total bit-rate is small, the optimal tradeoff may involve large motion errors, but this only occurs when the frames are very heavily quantized.

Quantization increases the range of motion error for which the linear relationship remains accurate. Recall that our linear approximation in equation (3) becomes invalid when the product of \(\omega\) and \(\delta\) becomes large. According to equation (4), for any particular values of \(\delta\) and \(\omega\), whose product is large, the non-linearity only has an effect if \(S_x(\omega) > 0\). Therefore, as high frequencies are removed by quantization, the linear approximation remains true for higher values of \(\delta\). The same effect is observed when high frequency components are discarded for reduced resolution reconstructions. As a result, it turns out that the optimal allocation of motion and sample data invariably occurs at combinations of motion and sample bit-rates for which the frame error is linearly related to the motion error.

The effect of frame sample quantization on motion sensitivity, for the first frame of the “Mobile & Calendar” sequence, is shown in Figure 1. The continuous curves represent the motion-induced frame distortion, obtained by averaging over many error orientations, as the error magnitude \(\|\delta\|\) is varied. In this case, the frame has first been compressed and reconstructed at 0.2 and 0.1 bits per
Figure 1: Increase in frame distortion due to motion error, for various reconstruction bit-rates. Linear and quadratic models are compared with experimental results.

pixel using an implementation of the JPEG2000 still image compression standard. We also show the results obtained when the original frame is used. Figure 1 clearly demonstrates the reduced motion sensitivity and increased linearity resulting from quantization.

The dashed lines in Figure 1 show the distortion predicted by the linear model of equation (10). The slopes are determined by directly evaluating equations (6),(7) and (9). For comparison, we also show the predicted distortion using the model derived by including two terms of the Taylor series from equation (3). In this case, it can be shown that the distortion model is a quadratic in $\|\delta\|^2$, given by

$$D_y \approx \|\delta\|^2 \Psi_x + \|\delta\|^4 \left( \frac{\Phi_{1,x} + \Phi_{2,x}}{4} \right)$$

where the factors $\Phi_{1,z}$ and $\Phi_{1,z}$ are given by

$$\Phi_{1,z} = \frac{1}{(2\pi)^2} \int \int S_x(\omega) \omega_1^4 d\omega$$
$$\Phi_{2,z} = \frac{1}{(2\pi)^2} \int \int S_x(\omega) \omega_2^4 d\omega$$

According to Figure 1, the linear model gives a reasonable approximation to the actual motion error. The quadratic model suffers less from over-estimation of the reconstructed frame distortion, and follows the actual distortion curve up to larger values of motion error.

It is possible to derive increasingly accurate models for the motion-induced frame distortion by incorporating more terms of the Taylor series expansion for $|1 - e^{-j\omega^t \delta}|^2$. However, the resulting models are polynomials in $\|\delta\|^2$, so that the rate-allocation procedure becomes much more complex.
In addition, our experimental results indicate that for rate-allocation purposes, non-linear models provide virtually no improvement in performance. This suggests that rate-allocation generally tends to involve sample and motion bit-rate combinations for which the linear model is sufficiently accurate.

3.2 Extension to Non-constant Motion Error

In practice, we cannot expect the motion error to be constant; however, we can expect it to be a smooth function of spatial position. This is because the deformable mesh model itself smoothly interpolates the displacements at each node. In the absence of specific knowledge concerning spatial variations in $S_x(\omega)$, the frame distortion model in equation (10) may be reasonably replaced by

$$D_y \approx D_v \Psi_x$$

where $D_v$ represents the mean squared motion error magnitude, averaged over all samples in the frame.

The triangular mesh used in our experiments is constructed from triangular “patches” in the manner depicted in Figure 2. In the figure, nodes are positioned on a unit-spaced grid. To find the mean squared motion error $D_v$, we first deduce the total squared motion error, integrated over all of the unit-dimensioned triangular patches in the frame. We then divide by the total area of these patches. This area is simply equal to the total number of nodes, or equivalently, the total number of motion vectors which are subject to quantization error.

In Figure 2, the motion vectors associated with the vertices of triangle D are represented by $v_1, v_2$ and $v_3$. The mesh model represents the motion $v(a, b)$, at location $(a, b)$ inside triangle D, by the following affine expression.

$$v(a, b) = v_1(1 - a) + v_2(a - b) + v_3b$$

Now suppose that $v_1$ is subjected to an error $\Delta v_1$, while $v_2$ and $v_3$ are error free. The squared magnitude of the error introduced into $v(a, b)$ is $\|\Delta v_1\| \cdot (1 - a)^2$. Integrating over the triangular
patch $D$, we obtain the total contribution of $\Delta v_1$ to squared motion error within triangular patch $D$. Using $E_D$ to denote this contribution, we have

$$E_D = \|\Delta v_1\|^2 \int_0^1 \int_b^1 (1 - a)^2 da \, db = \frac{\|\Delta v_1\|^2}{12}$$

The corresponding expressions for the remaining triangles shown in Figure 2, are

$$E_A = E_B = E_E = \frac{\|\Delta v_1\|^2}{12} \quad \text{and} \quad E_C = E_F = \frac{\|\Delta v_1\|^2}{6}$$

The total contribution of $\Delta v_1$ to squared motion error, integrated over the entire frame, is found by adding the contributions from each of these six triangles, yielding $E_1 = \frac{2}{3} \|\Delta v_1\|^2$.

For the mesh used in our experiments, every non-boundary node is connected to six triangles as in Figure 2. Assuming uncorrelated\(^1\), zero mean node vector errors, $\Delta v_n$, with magnitude variance $\sigma_n^2$, we find that the expected total squared motion error is given by $E = \frac{2}{3} \sum_n \sigma_n^2 \|\Delta v_n\|^2$. Dividing through by the number of node motion vectors, we find that the mean squared motion error magnitude $D_v = \frac{2}{3} D_W$, where $D_W$ denotes mean squared error in the node motion vector parameters. Equation (11) then becomes

$$D_y \approx \frac{2}{3} \Psi_x D_W \quad (12)$$

\(^1\)The assumption of uncorrelated quantization errors is commonly required in order to trace the impact of those errors through non-orthogonal linear systems. The assumption is most reasonable where the quantization errors are very small. In the worst case, if node vector errors turn out to have very high positive correlation, the factor of $2/3$ found here will approach 1.
Similar expressions may be obtained for other motion models. For example, typical block-based motion models involve a piecewise constant approximation of the underlying motion field. The motion field error is constant within each block, so $D_y \approx \Psi_x D_W$, which is 50% larger than that for the triangular mesh motion model.

### 3.3 Application to Temporal Transforms

We extend the model developed in the previous sections, for a single motion compensation operation, to the temporal synthesis system associated with the 5/3 incarnation of the LIMAT framework.

At each stage of the transform, the distortion introduced at each synthesis step is a combination of the distortion resulting from quantization in each warping operator. Assuming uncorrelated errors, equations (1) and (12) yield

\[
D_{2k} \approx \frac{1}{24} \left( \Psi_{h_{k-1}} D_{W_{2k-1-2k}} + \Psi_{h_k} D_{W_{2k+1-2k}} \right)
\]

\[
D_{2k+1} \approx \frac{1}{6} \left( \Psi_{x_{2k}} D_{W_{2k-2k+1}} + \Psi_{x_{2k+2}} D_{W_{2k+2-2k+1}} \right)
\] (13)

The update step maps high-pass subband frames onto the low-pass subband frames. High-pass subband frames contain temporal details not captured by the motion-compensated prediction step. Assuming the motion is reasonably well modeled, the energy in the high-pass frames is much less than the energy in the low-pass frames. This means that $\Psi_{h_k} \ll \Psi_{x_k}$, allowing us to ignore the contribution $D_{2k}$ to the total video distortion. In addition, accurate rate-allocation only becomes important at low bit-rates, where the power spectra of the high-pass subband frames are strongly attenuated by sample quantization, and $\Psi_{h_k}$ is particularly small.

Equations (13) refer to a single stage of temporal transformation. In general, we have a wavelet pyramid consisting of $T$ stages. Quantization errors in the motion fields used by the $t^{th}$ stage in the transform yield distortion in the frames, $x_{2k+1}^{(t)}$, reconstructed at this stage. These frames pass through $t-1$ stages of temporal wavelet synthesis when reconstructing the final video sequence. Let $G_{t-1}$ denote the energy gain factor associated with these synthesis operations. In the case of the 5/3 transform, the low-pass synthesis vectors $g_t[n]$, associated with $t$ levels of temporal synthesis,
are sampled B-splines of order one. That is,

\[ g_t[n] = \begin{cases} 
  1 - \left| \frac{n}{2^t} \right| & |n| \leq 2^t \\
  0 & |n| > 2^t 
\end{cases} \]

and

\[ G_t = \sum_{n=-2^t}^{2^t} (g_t[n])^2 = 1 + 2 \sum_{n=1}^{2^t} \left(1 - \frac{n}{2^t}\right)^2 \]

The distortion contributions due to quantization in the motion mappings \( W_{2k\rightarrow2k+1}^{(t)} \) and \( W_{2k+2\rightarrow2k+1}^{(t)} \) are

\[ \frac{1}{6} G_{t-1} \Psi_{x_{2k}} D_{W_{2k\rightarrow2k+1}}^{(t)} \quad \text{and} \quad \frac{1}{6} G_{t-1} \Psi_{x_{2k+2}} D_{W_{2k+2\rightarrow2k+1}}^{(t)} \]

respectively. As mentioned in Section 2, these are the only motion mappings whose parameters are actually coded.

The total motion-induced reconstructed frame error, \( D_M^{\text{total}} \), is given by

\[ D_M^{\text{total}} \approx \frac{1}{6} \Psi \frac{1}{V} \sum_{i=1}^{T} G_{t-1} D_{M,t}^{\text{total}} \quad (14) \]

where \( D_{M,t}^{\text{total}} \) denotes the total motion parameter squared error for all motion fields at temporal depth \( t \), and each motion mapping has \( V \) motion vectors. In equation (14) we have temporarily assumed that the energy spectrum is identical for all original frames and low-pass subband frames, allowing us to replace \( \Psi_{x_{2k}}^{(t)} \) with a single motion sensitivity term \( \Psi \).

4 A Layered Framework for Joint Scaling of Motion Parameters and Subband Samples

According to equation (14), \( \Psi \) acts as a global scaling factor for the video distortion contributions associated with each motion mapping. This property motivates our approach to structuring and managing a scalable motion representation. In particular, it justifies the creation of a rate-distortion optimal layered motion representation that is independent of the representation used for spatio-temporal subband sample data. With layered motion parameter and sample data representations,
we can determine an optimal tradeoff between motion and sample bit-rates, after the content has been coded. It is helpful at this point to introduce our proposed layering scheme, deferring discussion of specific motion parameter coding techniques until Section 5.

We create separate layered representations for the sample data and motion information using the embedded block-coding and rate-distortion optimized layering techniques of EBCOT [6]. Following spatio-temporal decomposition of the sample data, the subbands are partitioned into blocks, each of which is coded into a finely embedded bit-stream using the fractional bit-plane coding technique of JPEG2000 [6]. Each embedded bit-stream may be viewed as a sequence of segments, such that truncation at the end of the first \( n \) segments results in a representation of length \( L_{bs} (n) \), with an associated contribution of \( D_{bs} (n) \) to the total distortion in the reconstructed sample data. A similar approach is taken with the motion parameters, where truncating the motion parameter code-block \( b_M \) at the end of the first \( n \) segments results in a representation of length \( L_{bm} (n) \), with an associated motion parameter distortion contribution of \( D_{bm} (n) \).

Assuming uncorrelated distortions, the total squared error \( D \) in the reconstructed video may be written as the sum of the weighted distortion contributions from all motion parameter and subband sample code-blocks.

\[
D \approx \Psi \sum_{b_M} G_{bm}^M D_{bm} (n_{bm}) + \sum_{b_S} G_{bs}^S D_{bs} (n_{bs}) \quad (15)
\]

In equation (15), \( G_{bs}^S \) represents the relevant synthesis energy gain factor for each block of transformed sample data. Similarly, \( G_{bm}^M \) represents the synthesis energy gain factor for the motion code-blocks, combined with the scaling factors \( \frac{1}{16} G_{t-1} \) from equation (14). The indices \( n_{bs} \) and \( n_{bm} \) identify the number of initial bit-stream segments selected from each transformed sample code-block \( b_S \) and each motion code-block \( b_M \), respectively. The total length of the compressed video representation may be written as

\[
L = \sum_{b_m} L_{bm} (n_{bm}) + \sum_{b_s} L_{bs} (n_{bs}) \quad (16)
\]

The layering scheme of EBCOT is used to create a collection of sample quality layers, such that the first \( q \) layers represent optimally truncated prefixes of each sample code-block, for each
$q = 1, 2, \ldots$. A full description of this layering scheme may be found in [6]. However, in an effort to make the present paper as self-contained as possible, we provide a brief description here. Associated with each sample quality layer $q$ is a distortion-length slope threshold $\lambda_q^S$. The first $q$ sample data layers together contain $n_{bs}^q$ segments from the embedded bit-stream of each transformed sample code-block $b_S$, where $n_{bs}^q$ is the largest truncation point for which $S_{bs} \left( n_{bs}^q \right) \geq \lambda_q^S$, and $S_{bs} \left( n \right)$ is the distortion length slope, given by

$$S_{bs} \left( n \right) = G_{bs} \frac{D_{bs} \left( n - 1 \right) - D_{bs} \left( n \right)}{L_{bs} \left( n \right) - L_{bs} \left( n - 1 \right)}$$

Strictly speaking, the above is correct only so long as the segment boundaries, $n$, are restricted to those which define the convex hull of the distortion-length characteristic for code-block $b_S$. Also, for $n = 0$ the slope is taken to be $S_{bs} \left( 0 \right) = \infty$. In summary, sample layer $q$ may be understood as the collection of all segments $n$ from spatio-temporal subband code-blocks $b_S$ for which

$$\lambda_{q-1}^S > S_{bs} \left( n \right) \geq \lambda_q^S \quad (17)$$

A similar approach is taken with the motion parameters, resulting in a collection of motion quality layers, indexed by $l = 1, 2, \ldots$, with distortion-length slope thresholds, $\lambda_l^M$. The first $l$ motion layers together contain the initial $n_{bm}^l$ segments from the embedded bit-stream of each motion code-block $b_M$, where $n_{bm}^l$ is the largest truncation point such that $S_{bm} \left( n_{bm}^l \right) \geq \lambda_l^M$, and $S_{bm} \left( n \right)$ is the distortion length slope

$$S_{bm} \left( n \right) = G_{bm} \frac{D_{bm} \left( n - 1 \right) - D_{bm} \left( n \right)}{L_{bm} \left( n \right) - L_{bm} \left( n - 1 \right)}$$

Thus, motion layer $l$ consists of all segments $n$ from motion parameter code-blocks $b_M$ such that

$$\lambda_{l-1}^M > S_{bm} \left( n \right) \geq \lambda_l^M \quad (18)$$

Now consider the problem of minimizing the overall reconstructed video distortion $D$, given by equation (15), subject to a constraint on the overall compressed length $L$, given by equation (16). Equivalently, we wish to minimize $D + \lambda L$, finding the smallest parameter $\lambda > 0$ for which
$L$ satisfies the relevant length constraint. Ideally, therefore, we should include the first $n_{bM}^{(λ)}$ bit-stream segments from each motion code-block $b_M$ and the first $n_{bS}^{(λ)}$ bit-stream segments from each subband sample code-block $b_S$, where

$$n_{bM}^{(λ)} = \max \{ n \mid \Psi S_{bM} (n) \geq \lambda \} \quad \text{and} \quad n_{bS}^{(λ)} = \max \{ n \mid S_{bS} (n) \geq \lambda \}$$

(19)

If $q$ and $l$ can be found such that $\Psi \lambda^M_l = \lambda^S_q = \lambda$ then the relevant code-block contributions can be obtained simply by combining the first $q$ sample quality layers with the first $l$ motion quality layers.

One of the chief benefits of motion and sample quality layers is that they obviate the need to retain explicit knowledge of the distortion-length slopes $S_{bM} (n)$ and $S_{bS} (n)$ for each code-block truncation point. This slope information is approximated implicitly through the association of code-block bit-stream segments with layers and through the bounds expressed in equations (17) and (18). The tradeoff between the accuracy of this approximation and the overhead associated with signalling the code-block contributions to each layer may be controlled by selecting the separation between layer thresholds $\lambda^M_l$ and $\lambda^S_q$. In practice, to satisfy a compressed length constraint $L_{\max}$, we find a whole number of motion layers $l$ and sample layers $q$, with $\Psi \lambda^M_l \approx \lambda^S_q$, such that the total compressed length associated with these motion and sample layers does not exceed $L_{\max}$. The difference between the actual length of these layers and $L_{\max}$ is then filled in by including as much of sample layer $q + 1$ as possible.

Changes in the value of the motion sensitivity parameter $\Psi$ clearly affect the optimal tradeoff between motion and sample data. However, changes in the value of $\Psi$ do not affect the rate-distortion optimality of the motion layers. This means that a single set of motion layers can be used to jointly optimize the distribution of motion and sample data for a variety of different values of $\Psi$. As already noted in Section 3.1, we can expect $\Psi$ to exhibit a strong dependence on the spatial resolution at which we are interested in reconstructing the video.

In the above discussion we have ignored the dependence of $\Psi$ on the sample truncation points $n_S$, which do generally affect the energy spectra of the reconstructed video sequence. We rely on
the fact that the change in $\Psi$, scaled by $\sum_{b_{5m}} G_{b_{5m}}^M D_{b_{5m}} (n_{b_{5m}})$, resulting from a small increase in $n_{b_{5}}$, is generally much smaller than the corresponding change in $\sum_{b_{5}} G_{b_{5}}^S D_{b_{5}} (n_{b_{5}})$. In any event, for practical reasons, we are prevented from considering anything other than the sample distortion-length slopes $S_{b_{5}} (n)$ when constructing subband sample layers $q$. To account for the variation in $\Psi$ over a large range of reconstructed bit-rates, however, we estimate a separate motion sensitivity factor for each sample layer. This is discussed further in Section 6.2.

In order to account for changes in the video power spectra over time, we periodically update our estimates for the motion sensitivity factors, $\Psi_q$. In the proposed video coder, both the temporal subband frames and the motion fields are partitioned into epochs, known as “frame slots”. We currently estimate a single motion sensitivity for each layer $q$ in each frame slot, although this policy could conceivably be adjusted to estimate motion sensitivities in a manner which accounts for variations in scene activity, both spatially and temporally.

We have already noted that the motion sensitivity factor $\Psi$ depends significantly on the reconstructed spatial resolution, so the relative allocation of motion and sample bits also varies at each spatial resolution. Indeed we provide concrete evidence for this dependence in Section 5.2. By contrast with spatial resolution, the optimal rate-allocation is substantially insensitive to temporal resolution. Reconstruction at reduced temporal resolutions implies a scaling of the energy gain factors used to optimally allocate both sample and motion bits among temporal subbands, following equation (15). This scaling is approximately the same across temporal subbands, so the optimal distribution of motion and sample bits is largely unaffected by temporal resolution. Here we are assuming the power spectrum of the subband frames is reasonably independent of temporal depth, which is usually the case so long as the motion is effectively modelled.

5 Scalable Coding Techniques for Motion Parameters

The motion parameters corresponding to operators $\mathcal{W}_{2k \rightarrow 2k+1}^{(t)}$ and $\mathcal{W}_{2k+2 \rightarrow 2k+1}^{(t)}$, at each temporal depth $t$, are subjected to a reversible wavelet transform, prior to embedded coding of the motion
subbands. As mentioned in the previous section, we use the embedded coding and rate-distortion optimized layering techniques of EBCOT [6]. We apply fractional bit-plane coding to each motion subband. Note that the subbands could first be divided into blocks, as in EBCOT, if the number of motion parameters were large enough to justify it.

The embedded bit-streams may be truncated at any of a large number of coding pass boundaries, each of which has an associated code length $L_{bpM}(n)$, and an associated contribution $D_{bpM}(n)$, to distortion in the reconstructed video sequence. The distortion contributions are calculated by using the energy gain factors $G_{pm}^M$, which depend on the synthesis operations required to reconstruct the motion parameters from their subbands. Note that the true energy weight from equation (15) is $\Psi G_{pm}^M$, but the global scaling factor $\Psi$ has no effect on the motion layering. The length and distortion information is used to determine the number of incremental coding passes which should be contributed by each motion subband, at each temporal depth, to each successive motion layer.

We currently divide each bit-plane into three coding passes. Following the fractional bit-plane paradigm in [6, 10], context modeling is used both to determine coding pass membership and to code the significance information in each subband. As in other bit-plane coding schemes, the term “significance” is used to identify whether or not a transformed motion parameter has been found to be non-zero in any bit-plane encountered so far. We currently code the horizontal and vertical components of the reversibly transformed motion vector field together, using four significance probability modeling contexts for each direction. These contexts are based on the significance of spatially and temporally neighbouring motion parameters in the same and the orthogonal direction. JPEG2000’s MQ coder [6] is used for probability modeling and arithmetic coding of the bit-plane information.

While the embedded coding scheme is of some interest, the reversible transformation steps, which we apply to the motion parameters prior to coding, are of significantly greater interest. The choice of transform affects both the coding efficiency and the nature of the distortion introduced into the motion parameters when the subband bit-streams are truncated. We consider two types of
transformation steps, each of which employs reversible lifting [11], so that the motion parameters may be recovered losslessly if the embedded bit-streams are not truncated.

The first transform operates in the temporal dimension, on the node displacement vectors, \( \mathbf{v}^{(t)}_{k,f} [j] \) and \( \mathbf{v}^{(t)}_{k,b} [j] \), associated with each pair of motion maps, \( \mathcal{W}^{(t)}_{2k-2k+1} \) and \( \mathcal{W}^{(t)}_{2k+2-2k+1} \). These displacement vectors point respectively forward and backward to each node location, \( n_j \), on the same target frame, \( x^{(t)}_{2k+1} [n] \). Consequently, we expect \( \mathbf{v}^{(t)}_{k,f} [j] \approx -\mathbf{v}^{(t)}_{k,b} [j] \). The S-transform [12] (reversible Haar wavelet) is applied to \( \mathbf{v}^{(t)}_{k,f} [j] \) and \( -\mathbf{v}^{(t)}_{k,b} [j] \), yielding “temporally” low- and high-pass vector fields.

The second transform considered here operates in the spatial dimension. Specifically, we employ a reversible spatial DWT, based on the 5/3 wavelet kernel, extended to two levels of decomposition. This transform is applied separately to the horizontal and vertical components of each field of motion parameters or, if the S-transform is also employed, to the horizontal and vertical components of each of the temporally low- and high-pass vector fields.

The reason for selecting reversible, rather than irreversible transforms for the motion parameters is that we wish to preserve the ability to use the original unquantized motion parameters recovered directly from motion estimation, at both the encoder and the decoder. At higher compressed bit-rates, the best performance generally results from using all motion information recovered from the motion estimation stage.

### 5.1 Impact of Motion Parameter Transforms

To evaluate the impact of motion parameter transforms, we losslessly compress the original video sequence using a reversible lifting implementation of the motion compensated temporal transform, extended to \( T = 3 \) levels, with motion parameters estimated to \( 1/8^{th} \) pixel accuracy. The motion parameters are transformed and coded to produce 12 motion quality layers, having bit-rates from 5 kbps up to approximately 220 kbps, depending on the transformations used. The motion information is truncated to each layer boundary prior to reconstruction and the reconstructed video distortion is plotted as a function of the motion bit-rate. The distortion in this case is due exclu-
As mentioned above, the motion parameter transforms also affect the nature of motion quantization errors. In fact, the use of motion transformations can actually reduce the video distortion associated with a given level of motion MSE. The left plot in Figure 3 shows the video distortion plotted against motion MSE, for the same test conditions described above. Evidently, the use of transformations can increase the video PSNR by as much as 1.5 dB, for a given level of motion error. Similar results were observed using other test sequences.

Clearly, spatial transformation plays a particularly important role in minimizing the impact of motion quantization. The spatial transform also tends to concentrate motion errors in regions of highly non-uniform motion. These may also correspond to regions where the motion cannot be modelled accurately, in which case the low-pass subband frames will tend to be locally blurred by the imperfectly compensated temporal transform. This, in turn, tends to reduce the impact of motion error on the reconstructed video distortion, since blurred regions have less high spatial frequency content.
5.2 Validation of the Linear Model

At this point, it is interesting to investigate the relationship between motion parameter MSE and induced reconstructed video distortion more closely, to see under what conditions this relationship can be considered linear. The experimental observations reported in the left hand plot of Figure 4 are obtained from a single frame slot (32 frames in this case) of the CIF resolution “Foreman” sequence. The motion information is compressed using the methods described above, with both the spatial and temporal motion parameter transforms. The motion-compensated temporal subbands produced by a $T = 3$ level temporal transform are subjected to spatial transformation and scalable coding, using methods derived from the JPEG2000 standard.

We use 22 layers for the motion information, having cumulative bit-rates spaced logarithmically from approximately 4 kbps up to fully lossless reconstruction at 180 kbps. Note that fewer motion layers would normally be used in practice, since the signalling overhead required for many layers can affect coding efficiency. However, the cost of motion has no bearing on this particular experiment. At each subband sample bit-rate, the entire video sequence is reconstructed after truncating the motion information to each of the layer boundaries in turn. Since the cost of motion is not included in the bit-rate reported here, the increase in video distortion corresponding to each motion layer is purely the result of motion error. In all other respects, the experimental conditions used here are identical to those identified in Section 7.

The left plot in Figure 4 reveals a motion-induced distortion behaviour for the entire motion-compensated transform which is similar to that observed for a single motion compensation operation (see Figure 1). The curves are linear for small motion errors, with the linear region extending to higher motion errors when the sample information is quantized. Similar results are observed for other test sequences. For sequences with very smooth motion, such as “Mobile & Calendar”, the motion information is so compressible that near-linear behaviour is evident for virtually all interesting values of motion MSE.

The right-hand plot in Figure 4 shows the results obtained when the video is reconstructed at
Figure 4: Shows the effect of motion error on total squared error for a frame slot reconstructed at CIF (left) and QCIF (right) resolutions, at various bit-rates, using the Foreman sequence.

QCIF resolution, rather than its full CIF resolution. In this case, the relationship between motion error and video distortion is virtually linear for all motion errors. As mentioned previously, this is due to the significant reduction of high frequency energy in reduced resolution video reconstructions. An important question which arises in connection with reduced spatial resolution reconstructions is how distortion should be assessed. Although spatial scalability may be introduced into the LIMAT framework in a number of ways, for these experiments we adopt a simple approach that simply involves discarding high-pass spatial subbands prior to reconstruction. The reduced resolution distortion is computed by comparing the reconstructed sequence with that obtained when the motion and sample data are losslessly decoded. Note also that the full motion field resolution is used for reconstruction at all spatial resolutions.

6 Rate-Allocation Strategies

In this section we propose two methods to determine the optimal allocation of motion and sample data for each reconstructed bit-rate and spatial resolution. We first describe a straightforward, search-based rate-allocation strategy. While somewhat time-consuming, this method is guaranteed to find the most efficient balance between sample and motion bit-rates. A more computationally efficient approach is described in Section 6.2. This scheme explicitly utilizes the linear model
developed in Section 3, which provides a mechanism for directly quantifying the effect of motion error prior to reconstruction. We find that this scheme performs comparably to the search-based approach, which reinforces the validity of the linear model itself.

Both rate allocation schemes effectively produce tables with one entry for each motion quality layer, identifying the range of bit-rates (or sample quality layers) for which that layer is the optimal place at which to truncate the motion representation. One such table is produced for each spatial resolution of interest, in each frame slot. We do not count the cost of storing these tables in any of our numerical experiments, since in many cases the tables would be used only by an intermediate server or transcoder. However, the tables are not large and could be efficiently compressed. This is because most of the information required to extract rate-distortion optimized subsets of the video bit-stream is already embodied within the association between code-block bit-stream segments and their respective motion and sample quality layers. The cost of coding this layer association information (known as Tier 2 information in EBCOT [6]) is included in all our experiments.

6.1 Brute Force Search Method

In this method, the optimal balance between motion and sample bit-rates is determined for each reconstructed video bit-rate of interest, by decoding the entire video sequence multiple times, each time truncating the motion representation to a different number of layers. The compressor builds a table identifying the optimal number of motion layers to use for each compressed video bit-rate. Since motion sensitivity depends on the spatial resolution, a separate table is constructed for every desired reconstructed spatial resolution. An example of such a table is given in Section 7.

The brute force scheme represents a robust approach to rate-allocation. An obvious disadvantage with this scheme is the large amount of computation involved; however, the computation is not as bad as might at first be supposed. Firstly, since the optimal number of motion layers varies monotonically with the overall bit-rate, and there are only a limited number of motion layers, it is usually sufficient to perform the optimization experiment over a relatively small set of bit-rates. The monotonicity of the optimal motion layer vs. bit-rate function may be further exploited to
reduce the set of motion layers for which the video must be decompressed. In particular, having found the optimal number of motion layers \( l_R \), corresponding to overall bit-rate \( R \), for the next higher bit-rate \( R' \) it is sufficient to try \( l_{R'} = l_R + k \) for each successive \( k = 0, 1, 2, \ldots \), until the reconstructed video ceases to decrease. Some thought should reveal that this limits the total number of decompression attempts to two per test bit-rate, plus one per motion quality layer.

### 6.2 Model-based Rate-Allocation with the Spatial DWT

The linear motion distortion model proposed in Section 3 not only justifies the creation of independent motion and sample quality layers, but also facilitates a computationally efficient approach to rate-allocation. Unlike the brute force method, the rate-allocation scheme presented here relies on explicitly evaluating the motion-distortion model, following estimation of the motion sensitivity parameters \( \Psi \).

In every frame slot we compute a motion sensitivity factor \( \Psi \), for each sample quality layer \( q \), and for each desired reconstructed spatial resolution. As mentioned in Section 4, \( \Psi \) acts as a global scaling factor for the distortion-length slopes associated with the embedded code-block bit-streams used to represent motion parameters. Ideally, all of the motion code-blocks \( b_M \) and subband sample code-blocks \( b_S \) would be truncated to points \( n_{bM} \) and \( n_{bS} \), whose distortion-length slopes, \( S_{bM}(n_{bM}) \) and \( S_{bS}(n_{bS}) \), are as similar as possible, following equation (19). We noted that this is equivalent to truncating the layered motion and sample data representations at layer boundaries \( q \) and \( l \), respectively, so long as their distortion-length slope thresholds satisfy \( \Psi \lambda^M_q = \lambda^S_q \) and these layer boundaries yield the desired overall compressed video bit-rate.

As with the brute force rate-allocation scheme, we restrict our selection of motion bit-rates to those corresponding to whole motion layers. This reduces the amount of distortion-length slope information that must be available for rate-allocation, and guarantees that the reconstructed motion information is optimally distributed amongst the temporal resolutions and motion mappings in any given frame slot.

To estimate \( \Psi \), we evaluate equations (6) and (7), which requires an estimate of the video power
spectral density (PSD). Common methods for power spectrum estimation generally involve either computing averaged Periodograms or finding the Fourier Transform of a windowed autocorrelation estimate [13]. It is straightforward to use techniques such as these to find the spectrum of the original video and also reduced resolution versions of the original video. However, with these approaches, there is no simple way to account for the effects of subband sample quantization on the reconstructed power spectra. In addition, spectral estimation of video data can be very computationally demanding, and computational efficiency is the primary motivation for seeking an alternative to the brute force scheme advanced in Section 6.1.

We propose an alternative approach that avoids both of the drawbacks mentioned above. We obtain a PSD estimate from frequency information inherent in the spatio-temporal wavelet decomposition. Since the DWT is already part of the compression process, the proposed method incurs very little additional computational cost. More importantly, we can easily utilize the same information that is collected for the rate-distortion optimized layering process, to determine the effect of quantization on the reconstructed video power spectra.

In the EBCOT algorithm, each spatio-temporal subband is partitioned into code-blocks which are coded separately using a fractional bit-plane coding algorithm. Recall that the resulting embedded bit-streams consist of a large number of segments, each of which has an associated code length $L_{bS}(n)$, and an associated contribution $D_{bS}(n)$, to distortion in the reconstructed video sequence.

For a particular sample quality layer $q$, the rate-distortion optimal layering algorithm finds the most appropriate truncation point $n_{bS}^q$ for each code-block $bS$. We can also approximate the energy in the code-block, if only the first $q$ sample layers are received, by $D_{bS} \left( n_{bS}^q \right)$. To understand this approximation, let $X$ denote the random vector representing all samples in code-block $bS$ and let $\hat{X}_n$ denote the random vector representing the quantized samples associated with truncation point $n$. Note also that $X$ has zero mean for all but the lowest frequency spatio-temporal subband – that subband is of no interest to us, as $\Psi$ is insensitive to very low spatial frequencies. Thus, $\|X\|^2$ is the unquantized energy of the code-block. A well-known property of any optimal quantizer$^2$ is that

---

$^2$This property holds for any quantizer whose representation vectors (levels, for a scalar quantizer) are the centroids
the quantization error, $\Delta_n = X - \hat{X}_n$, is uncorrelated with $\hat{X}_n$. It follows that

$$E[\|\hat{X}_n\|^2] = E[\|X\|^2] - E[\|\Delta_n\|^2] = E[D_{bs}(n)]$$

where we have used the fact that $D_{bs}(n)$ is the difference between the distortion when no information is available for the code-block (i.e., $\|X\|^2$) and the distortion when the block bit-stream is truncated after the first $n$ segments (i.e., $\|\Delta_n\|^2$). Of course, the quantization process associated with embedded block coding might not be optimal, and even if it were, the equality between $\|\hat{X}_n\|^2$ and $D_{bs}(n)$ holds only for statistical averages. For these reasons, the energy of the quantized code-block in layer $q$ will only be approximately equal to $D_{bs}\left(n^{q}_{bs}\right)$, but we find this approximation to be quite acceptable and highly convenient in practice.

Adding the energy contributions from all code-blocks in a particular spatial subband gives us an estimate of the total energy in the corresponding spatial frequency band. By repeating this for all spatial subbands, and all quality layers, we build estimates of the video power spectrum at each reconstructed quality layer. Although the optimal video sample bit-rate may not correspond exactly to a whole sample quality layer, the motion sensitivity changes only slowly between layers, so long as a reasonable number of layers are used.

To estimate the motion sensitivity at reduced spatial resolutions, we adopt the same procedure, ignoring the high-pass spatial subbands which are discarded prior to reconstruction.

Obtaining our spectral estimates with the DWT has several limitations. Firstly, the dyadic structure of the DWT limits the resolution at which spectral estimates are available. We currently assume that the spectral density is uniform within each spatial subband and that the subbands perfectly partition the spatial frequency spectrum. A second limitation is that the spatio-temporal DWT is not an orthogonal decomposition. This means that adding the energy in each subband, after adjusting for the appropriate synthesis energy gain factors, does not exactly produce the true synthesized video energy. To minimize this effect we prefer to employ the 9/7 wavelet transform spatial decompositions, since this transform is much closer to orthogonal than the 5/3.
Table 1:  Effect of motion bit-rate on video PSNR. Results for proposed coder is shown above those for non-scalable compression.

<table>
<thead>
<tr>
<th>Motion (kbps)</th>
<th>Total Bit-rate (kbps)</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>400</th>
<th>800</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>19.55</td>
<td>20.09</td>
<td>20.42</td>
<td>20.63</td>
<td>20.76</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20.18</td>
<td>21.34</td>
<td>22.71</td>
<td>24.49</td>
<td>26.93</td>
<td></td>
</tr>
<tr>
<td></td>
<td>20.27</td>
<td>21.71</td>
<td>23.34</td>
<td>25.21</td>
<td>27.70</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>19.80</td>
<td>21.58</td>
<td>22.94</td>
<td>23.92</td>
<td>24.47</td>
<td></td>
</tr>
<tr>
<td></td>
<td>19.88</td>
<td>21.93</td>
<td>23.95</td>
<td>26.05</td>
<td>28.63</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>18.63</td>
<td>21.46</td>
<td>23.69</td>
<td>25.37</td>
<td>26.80</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18.66</td>
<td>21.60</td>
<td>24.13</td>
<td>26.79</td>
<td>29.63</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>-</td>
<td>19.50</td>
<td>23.57</td>
<td>26.62</td>
<td>29.17</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-</td>
<td>19.50</td>
<td>23.67</td>
<td>27.01</td>
<td>30.39</td>
<td></td>
</tr>
<tr>
<td>215</td>
<td>-</td>
<td>-</td>
<td>25.12</td>
<td>30.12</td>
<td>30.12</td>
<td></td>
</tr>
</tbody>
</table>

Despite the limitations mentioned above, we find that the performance of this model-based scheme is generally comparable to that of the brute force scheme. Since rate-allocation only involves identification of the optimum motion layer, rather than an optimal bit-rate, a certain degree of inaccuracy in the estimated motion sensitivity can be tolerated. Experimental evidence for these assertions is provided in the next section.

7  Experimental Results

Results provided in this section are obtained using the “Foreman”, “Stefan”, “Mobile & Calendar” and “Bus” sequences. Each original sequence is CIF resolution, with a frame rate of 30 frames per second. The sequences are all 300 frames in length, except “Bus” which has 150 frames. For these results, the sample data is compressed using \( T = 3 \) stages of the motion-adaptive temporal transform. The subband frames are subjected to five levels of irreversible spatial DWT, followed by embedded block coding of the quantized wavelet coefficients, using a modified implementation of the JPEG2000 image compression standard. A constant number of bits is allocated to each quality layer of each frame slot, where the frame slots have a duration of 32 frames.

The deformable triangular mesh motion model uses a node spacing of 16 pixels\(^3\) and motion vectors are estimated to \(1/8^{th}\) pixel accuracy. Spatial interpolation is performed using a \(7 \times 7\)

\(^3\)In our experience, finer meshes (e.g. with a node spacing of 8 pixels) do not appear to be beneficial.
Figure 5: Rate-distortion performance of various rate-allocation strategies, for the Bus (left) and Stefan (right) sequences at CIF resolution reconstruction.

The interpolation kernel obtained by windowing cubic spline functions. The scalable motion coding scheme described in Section 5 is used to generate 12 motion quality layers for each frame slot. The highest layer represents lossless compression of the motion, while the remaining layers represent successively lower motion bit-rates, down to 5 kbps.

Table 1 shows an example of the table built by the brute force search method described in Section 6.1, in this case for full spatial resolution, using the “Stefan” sequence. For various motion layers and total compressed video bit-rates, two reconstructed luminance PSNR values are shown. The first (upper) value identifies the PSNR obtained when the video is reconstructed using that motion layer, having been compressed using the original estimated motion parameters. The second (lower) value identifies the PSNR obtained when the video is both compressed and decompressed using the same quantized motion parameters. Of course, this is not an option for applications requiring scalability, since the compressor must know the target bit-rate prior to generating the bit-stream. Thus, the difference between the first and second values, taken at their respective maxima, may be legitimately interpreted as the cost of scalability. In the table, the maxima of the first values are shown in bold, and the maxima corresponding to the second values are underlined.

Evidently, the cost of scalability is small, at no point exceeding 0.44 dB. This is because the optimal motion bit-rate is invariably a small fraction of the total compressed bit-rate. At non-
optimal combinations of the motion and sample bit-rates, the difference between the upper and lower values presented in the table can be much larger. The last row in the table identifies the performance which may be obtained if we only scale the coded sample data, sending all of the estimated motion information. Motion scalability clearly offers substantial benefits at lower bit-rates, especially when the cost of sending the original motion parameters approaches or exceeds the overall bit-rate of interest.

Figure 5 shows the rate-distortion performance of the proposed coding scheme, using the “Stefan” and “Bus” sequences, at full resolution reconstruction. For each sequence we compare the compression performance achieved using the brute force and model-based rate-allocation schemes. We also show, using dotted curves, the compression performance achieved when all the motion information is transmitted. The continuous curves represent the non-scalable compression performance, as described above. Figure 6 shows similar results for the “Foreman” and “Mobile & Calendar” sequences, except that in this case the video sequences are reconstructed at QCIF resolution, having been originally compressed at the full CIF resolution.

Evidently, in all cases the model-based rate-allocation scheme achieves virtually identical performance to the brute-force scheme. As mentioned previously, the performance of the model-based rate-allocation scheme serves to reinforce the validity of the linear model itself. Each figure also indicates that the cost of scalability is small using either rate-allocation method, and significant performance improvements are observed relative to sending all the motion information.

8 Conclusions

In the context of a wavelet-based scalable video coder, the video distortion introduced by scaling the motion information after compression is approximately linearly related to the motion parameter MSE. With an additive distortion model and a highly scalable motion parameter bit-stream, it is possible to find an optimal tradeoff between sample and motion data bit-rates. We propose two methods to do this. A search-based method is robust, yet more computationally demanding than a
second method based on explicitly estimating the effect of motion error on video distortion. With both proposed rate-allocation strategies the cost of introducing scalability is small. At low bit-rates, significant gains are observed relative to lossless coding of the motion information.

References


