Compressible Motion Fields

Giuseppe Ottaviano*
Università di Pisa
ottavian@di.unipi.it

Pushmeet Kohli
Microsoft Research Cambridge
pkohli@microsoft.com

Abstract

Traditional video compression methods obtain a compact representation for image frames by computing coarse motion fields defined on patches of pixels called blocks, in order to compensate for the motion in the scene across frames. This piecewise constant approximation makes the motion field efficiently encodable, but it introduces block artifacts in the warped image frame.

In this paper, we address the problem of estimating dense *motion fields that, while accurately predicting one frame from a given reference frame by warping it with the field, are also* compressible.

We introduce a representation for motion fields based on wavelet bases, and approximate the compressibility of their coefficients with a piecewise smooth surrogate function that yields an objective function similar to classical optical flow formulations. We then show how to quantize and encode such coefficients with adaptive precision.

We demonstrate the effectiveness of our approach by comparing its performance with a state-of-the-art wavelet video encoder. Experimental results on a number of standard flow and video datasets reveal that our method significantly outperforms both block-based and optical-flow-based motion compensation algorithms.

1. Introduction

Most modern video compression algorithms fall into the category of *hybrid* video encoders that work by using previously decoded frames and some *side information* (explicitly added by the encoder) to make a *prediction* for the current frame. The difference between the prediction and the frame, called *residual* or *prediction error*, is then encoded separately to correct the prediction. On one hand a better prediction implies a more compact residual encoding; on the other hand, the side information must be kept as compact as possible to avoid its encoding cost outweighing the benefits of the more accurate prediction.

The largest part of such side information consists of a *motion field*, which allows to compensate for the motion of the camera and the objects in the scene across consecutive frames, hence forming a *motion compensated* prediction. Given a pair of images I_0 and I_1, a (dense) *motion field* u is a field of per-pixel *motion vectors* describing how to warp pixels from I_1 to form a new image $I_1(u)$, which we refer to as I_1 *warped with* u. Such an image can be used directly as a prediction of I_0; the residual is then $I_0 - I_1(u)$

It should be noted that the side-information (motion field in this case) does not need to be an estimate of the true motion in the scene, as in optical flow problems; we just want a motion field which results a residual that is small to encode. Ideally, we would associate to each pixel in the image the motion vector that minimizes the residual; however such a field may contain more information than the image itself: a field for n pixels has $2n$ degrees of freedom. Hence some freedom in computing the field must be traded for efficient encodability.

The traditional and most successful solution for the above problem is the family of Block Motion Compensation (BMC) algorithms, originally introduced in [11] and adopted by virtually every modern video coding algorithm. In its basic version, a fixed block size (say 16×16) is chosen and a motion vector is associated to each block. In the above definition this is equivalent to requiring that the motion field is *constant* within the blocks. This piecewise constant flow approximation makes the motion field efficiently encodable, but it introduces block artifacts in the decoded image frame.

In this paper we address the problem of estimating dense motion fields which, while accurately predicting one frame from a given reference frame by warping it with the field, are also *compressible*. We introduce a new representation for motion fields as linear combinations of a given basis. The computation of the basis coefficients can be posed as a global piecewise-smooth optimization problem which resembles classical optical flow formulations, optimizing for both *compressibility* and residual magnitude. The real-valued solution is then quantized and encoded, with a quantization algorithm that minimizes the error induced in the warping.

*Part of the work done while the author was an intern at Microsoft Research Cambridge.

The basis must be chosen to be able to represent sparsely a wide variety of motions, and to allow efficient optimization. We focus on wavelet bases, which loosely generalize block-based algorithms, and whose orthogonality simplifies the optimization. We perform a thorough experimental evaluation of our method on the Middlebury optical flow image pairs as well as a variety of video sequences. Our results reveal that our wavelet motion fields outperform both block-based and optical-flow-based motion compensation techniques.

Our contributions We summarize here the contributions of our work: *(1)* we introduce a new representation of motion fields based on orthogonal wavelets; *(2)* we show how to compute a motion field in this representation while optimizing for both residual quality and compressibility of the field; *(3)* we show how to quantize and encode the coefficients minimizing the warping error introduced by the quantization.

2. Related Work

As mentioned before, block-based algorithms induce motion fields that are likely to be discontinuous at block boundaries, thus introducing in the predicted image, and in-turn in the residual, discontinuities that are visually noticeable and expensive to encode. To alleviate this problem, either a de-blocking filter is used for post-processing the result (as in H.264 [19]), or the blocks are allowed to overlap. The latter approach averages the pixels from different blocks on the overlapping area using a smooth window function. Such a solution is called Overlapping Block Motion Compensation (OBMC) [16, 18], and is used in Dirac [6] and other wavelet-based codecs. Both solutions reduce the block artifacts but introduce blurriness, thus losing detail.

Different block sizes in a block motion compensation algorithm give different accuracy/compactness trade-offs. To account for parts of the image where higher precision is needed, e.g. across object boundaries, Variable Block Motion Compensation (VBMC) was proposed in [3]. In VBMC, each block can be segmented into smaller sub-blocks, with the segmentation encoded as side information, and a different motion vector is encoded for each sub-block. This approach is used in most modern video coding algorithms, including H.264 and Dirac. In both BMC and VBMC, the computation of the field is a discrete optimization problem: shorter motion vectors can be encoded with fewer bits, so a trade-off between residual magnitude and encoding cost must be decided; furthermore, the decisions on a block influence the decisions on other blocks, because the motion vectors are encoded differentially. VBMC exacerbates the problem by adding the decision of the segmentation, as more refined segmentations require more bits. Given the intrinsically combinatorial nature of the problem, no efficient optimal algorithms have been devised. Hence, greedy strategies are used in practice.

A large amount of work has been done on exploring alternate representations for dense motion, but the improvements over block-based solutions have been so marginal that the increase in complexity is not justified. Here we report the papers that are closest to our work.

In [14] the authors present a mesh-based representation (similar to the one used in [10]). However, the model is not expressive enough to cover a wide range of motions. In fact, the authors suggest to fall back to block-based methods when the mesh-based approach fails. Compact representations of dense motion fields are explored in [12], where the authors use a DCT-based encoder for the field, in [15] where a multiscale approach similar to our wavelet decomposition is used, and in [8], which uses a quad-tree-like hierarchical representation.

The main difference between the aforementioned papers and our work is in the estimation of the motion field: they use either a quadratic penalty on the field derivatives (similar to the Horn-Schunck model [9]) or MRF formulations, in order to favor smooth solutions that approximate the actual motion of the scene. This approach is based on the reasonable assumption that smooth fields are easy to compress[1]. In what follows, we show that by using a penalty modeled after the actual entropy encoder of the field, it is possible to obtain fields that can be encoded in significantly fewer bits, compared to the fields obtained with the smoothness penalty, without sacrificing prediction quality.

3. Problem statement

Notation We define a single-component[2] (grayscale) image I of width w and height h as a vector in $\mathbb{R}^{w\times h}$, and a *motion field* u as a vector in $\mathbb{R}^{2\times w\times h}$, with u^0 and u^1 being respectively the horizontal and vertical components. For a motion field to be *feasible* we constrain its motion vectors inside the image rectangle, i.e. $0 \le i + u^0_{i,j} \le w-1$ and $0 \le j + u^1_{i,j} \le h-1$. We call the set of feasible fields $\mathcal{F}$.

By a slight abuse of notation we extend I to the continuous rectangle $[0, w-1]\times[0, h-1]$ by interpolation (in our implementation we use bicubic) and define the image I *warped* with the motion vector u as $I(u)$, formally $I(u)_{i,j} = I_{i+u^0_{i,j}, j+u^1_{i,j}}$ (for instance, $I(\mathbf{0}) = I$). This notation allows us to write the *residual* of I_0 and I_1 under the motion field u as $I_0 - I_1(u)$.

3.1. Representation and coding cost

Field representation We represent a motion field u by its coefficients $\boldsymbol{\alpha}$ in a linear basis represented by a matrix W, so that $u = W\boldsymbol{\alpha}$ and $\boldsymbol{\alpha} = W^{-1}u$. The coefficients $\boldsymbol{\alpha}$ are

[1] In fact, modern optical flow algorithms favor fields with sharp edges at object boundaries, which would be harder to compress.

[2] We focus on grayscale images for brevity, but everything can be easily generalized to color images.

lossily encoded using a quantizer and an entropy coder. This is not dissimilar to *lossy transform coding* for images, used for example in DCT coders such as JPEG and wavelet encoders such as JPEG2000. However we will use an ad-hoc quantizer, as described in Section 5.

Coding cost Let $\widetilde{\mathcal{F}}$ be the set of feasible fields with *integer coefficients* in the basis W, and let $\mathrm{bits}(W^{-1}\tilde{u})$ denote the *coding cost* of $\tilde{u} \in \widetilde{\mathcal{F}}$, i.e. the number of bits obtained by coding the coefficients of $W^{-1}\tilde{u}$ with an entropy encoder. Given a bit budget B for the field, we wish to minimize the residual subject to the budget

$$\min_{\tilde{u}\in\widetilde{\mathcal{F}}} \|I_0 - I_1(\tilde{u})\| \quad \text{s.t. } \mathrm{bits}(W^{-1}\tilde{u}) \leq B \tag{3.1}$$

where $\|\cdot\|$ is a distortion measure such as L_1 or L_2^2.

Following the approach of Rate-Distortion Optimization [17] we can rewrite (3.1) as the Lagrangian

$$\min_{\tilde{u}\in\widetilde{\mathcal{F}}} \|I_0 - I_1(\tilde{u})\| + \lambda\, \mathrm{bits}(W^{-1}\tilde{u}) \tag{3.2}$$

where λ is the Lagrangian multiplier, which trades off bits of the field encoding for residual magnitude. This parameter can be either set a priori (estimating it from the desired bitrate) or optimized. The $\mathrm{bits}(\cdot)$ function is in general a complex algorithm, making the integer program (3.2) intractable. In order to optimize it we will derive a tractable surrogate function in Section 4.

3.2. Wavelet field representation

In the following we will assume that W is a block-diagonal matrix $\mathrm{diag}(W', W')$, i.e. the horizontal and vertical components of the field are transformed independently with the same transform matrix, and that W' is an orthogonal separable multilevel wavelet transform, so we can write $W^{-1} = W^T$. The coefficients of $W^T u$ can be divided into ℓ *levels*, which represent the detail at each level of the recursive wavelet decomposition, and in the separable 2D case each level (except the first) can be further divided into 3 *subbands*, which correspond to horizontal, vertical and diagonal detail. A comprehensive account of multilevel wavelet decompositions can be found in [13]. We denote the b-th subband as $(W^T u)_b$, and its i-th coefficient is $(W^T u)_{b,i}$.

VBMC and the Haar Wavelet Here we show that the motion fields obtained with Variable-Block Motion Compensation can be represented sparsely in the Haar wavelet basis, suggesting that other wavelet bases may be good choice for representing motion as well.

We assume that the blocks have all power-of-two sides; virtually all implementations of VBMC have this property. In the first approximation level of the Haar decomposition, each coefficient corresponds to the average of the motion field inside a macroblock (modulo normalization constants). Each level of splitting of the blocks adds a constant number of coefficients to the corresponding detail level in the Haar decomposition, which correspond to the difference of the vectors in the sub-blocks. As a consequence the total number of non-zeros is linear in the number of sub-blocks.

Where VBMC and the Haar representation differ is in the encoding of both the coefficients and the topology of the segmentation. VBMC represents the segmentation explicitly, and encodes the difference of the vector with the median of the neighboring motion vectors, to exploit the local coherency of the field. In the wavelet representation the segmentation is implicitly encoded in the set of non-zeros of the coefficients, while the local coherency is exploited by the recursive encoding of averages and differences: if neighboring blocks have similar field, the difference coefficient will be small. This is the same principle that makes wavelet bases suitable to represent natural images.

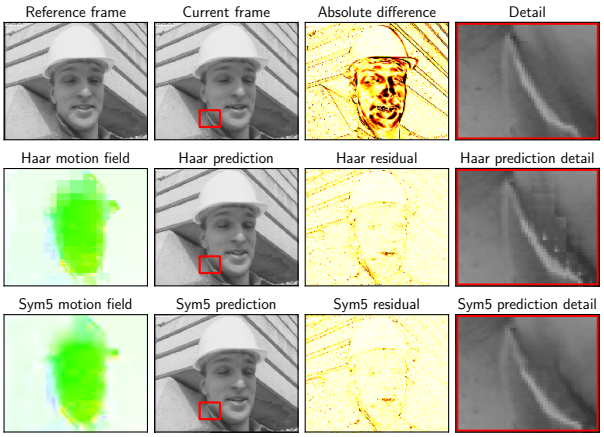

Figure 1. First row: first two frames of foreman and absolute difference. Second and third row: motion fields, prediction and residual obtained with Haar and Sym5 wavelets. One can observe that the Sym5 field is free from blocking artifacts.

4. Objective function derivation

Surrogate coding cost In this section we derive a tractable surrogate of $\mathrm{bits}(\cdot)$. We first start by noting that to encode the coefficients of $W^T u$ we need to encode both the *positions* of the non-zero coefficients (this term is significant if the representation is sparse) and the *sign and magnitude* of the quantized coefficients. Let $\tilde{u}$ be a solution of (3.2) with *integer* coefficients in the transformed basis, hence already quantized, and let n_b be the number of coefficients in the subband b and m_b the number of non-zeros.The entropy of the set of positions of the non-zeros in a given subband can be upper bounded by $m_b(2 + \log(\frac{n_b}{m_b}))$ by using a standard upper bound on the entropy [4]. Hence the contribution of each coefficient $\tilde{\alpha}_{b,i} = (W^T\tilde{u})_{b,i}$ can be written as $(\log n_b - \log m_b + 2)\mathbb{I}[\alpha_{b,i} \neq 0]$. Since optimizing over the

sparsity of a vector is a hard combinatorial problem, we make two approximations: first, we fix m_b to a small constant, assuming sparsity of the solution. Second, we approximate the indicator function $\mathbb{I}[\alpha_{b,i} \neq 0]$ with $\log(|\alpha_{b,i}| + 1)$. To approximate the non-zero coefficients cost we assume that the number of bits needed to encode a coefficient α can be bounded by $\gamma_1 \log(|\alpha| + 1) + \gamma_2$; this is true for many universal codes for integers, such as the Gamma codes [7] used in our entropy encoder.

Putting together the two approximate costs, we approximate the per-coefficient surrogate bit cost with $(\log n_b + c_{b,1}) \log(|\alpha_{b,i}| + 1) + c_{b,2}$, with $c_{b,1}$ and $c_{b,2}$ constants. By writing $\beta_b = \log n_b + c_{b,1}$ and ignoring $c_{b,2}$ we can define the *surrogate coding cost*

$$\|W^T u\|_{\log,\beta} = \sum_b \beta_b \sum_i \log(|(W^T u)_{b,i}| + 1). \quad (4.1)$$

Substituting this in (3.2) we obtain our final objective

$$\min_{u \in \mathcal{F}} \|I_0 - I_1(u)\| + \lambda \|W^T u\|_{\log,\beta}. \quad (4.2)$$

Despite the somewhat involved derivation, equation (4.2) is very simple, and reminiscent of the classical Horn-Schunck model for optical flow [9]. Instead of using a regularizer based on the *derivative* of the field, our formulation adopts a weighted logarithmic penalty on the *transformed coefficients*. Logarithmic (and in general concave) penalties are known to encourage sparse solutions [2]; in fact the motion fields we obtain have very few non-zero coefficients. This gives an intuitive explanation of why the resulting fields are compressible.

Additional sparsity can be enforced by controlling the parameters β_b; for instance, β_b can be set to ∞ to constrain the b-th subband to be zero. This can be useful if we want to obtain a locally constant motion field, by discarding the higher-resolution subbands. In the Haar case discarding the last two levels of the wavelet decomposition is equivalent to imposing a minimum block size of 4×4.

Data term linearization To optimize the objective function we follow the same strategy used in most optical flow algorithms: to handle the highly non-linear data term, we linearize it and iteratively solve the problem by refining the linearization at each iteration.

Given a field estimate u_0 we perform a first-order Taylor expansion of $I_1(u)$ at u_0, giving a linearized data term $\|I_0 - (I_1(u_0) + \nabla I_1[u_0](u - u_0))\|$ where $\nabla I_1[u_0]$ is the image gradient of I_1 evaluated at u_0. Rewriting the term as $\|\nabla I_1[u_0]u - \rho\|$ with ρ a constant term, the linearized objective is

$$\|\nabla I_1[u_0]u - \rho\| + \lambda \|W^T u\|_{\log,\beta}. \quad (4.3)$$

This function is a good approximation only when u is very close to u_0. A common solution to account for large displacements is embedding the linearized objective function in a coarse-to-fine manner; however, in our experiments this technique failed to find good solutions. The solution we adopted is to bootstrap the algorithm with an optical flow computed with Horn-Schunck (which is implemented in a coarse-to-fine manner). We compute a small set of optical flow solutions with different regularization parameters, and choose the one that minimizes the objective function in (4.2). Such flow is then used to initialize the iterative linearization.

Optimization of the linearized objective The function (4.3) is non-convex and hard to solve in general. However the two terms constituting it are easy to handle individually. For this reason, we use a decomposition based approach used in [21]. Specifically, we decompose the problem by introducing an auxiliary variable v and a quadratic coupling term that keeps u and v close as

$$\|\nabla I_1[u_0]v - \rho\| + \frac{1}{2\theta}\|v - u\|_2^2 + \lambda \|W^T u\|_{\log,\beta}. \quad (4.4)$$

The objective (4.4) can be minimized by alternating optimization, letting the coupling parameter θ decrease at each iteration (we adopt an exponentially decreasing schedule). We also project u to the rectangle $\mathcal{F} \cap [u_0 - \mathbf{1}, u_0 + \mathbf{1}]$ at each iteration, so that each step cannot update the field by more than one pixel (after which the linearization becomes inaccurate).

Keeping u fixed, $\|\nabla I_1[u_0]v - \rho\| + \frac{1}{2\theta}\|v - u\|_2^2$ can be optimized over v in closed form for both L_1 and L_2^2 norms, in the first case by soft-thresholding the field as described in [21], in the second case by solving a 2×2 linear system for each pixel.

Keeping v fixed, we show how to optimize $\frac{1}{2\theta}\|v - u\|_2^2 + \lambda \|W^T u\|_{\log,\beta}$ over u. Note that by the change of variable $z = W^T u$, the function becomes $\frac{1}{2\theta}\|W(W^T v - z)\|_2^2 + \lambda \|z\|_{\log,\beta}$. Since W is orthogonal, this is equal to $\frac{1}{2\theta}\|W^T v - z\|_2^2 + \lambda \|z\|_{\log,\beta}$. The problem is now separable, hence it can be reduced to component-wise optimization of the one-dimensional problem $(x - y)^2 + t \log(|x| + 1)$ in x for a fixed y. It can be easily seen that the minimum is either 0 or $\frac{1}{2}\operatorname{sgn}(y)(y - 1 + \sqrt{(y+1)^2 - 4t})$ (when the latter exists), so both points can be evaluated to find the global minimum.

5. Quantization

The solution u to (4.2) is real-valued; we now need to encode it lossily into a finite (possibly small) number of bits while not degrading too much the quality of the residual. To do this we follow the standard approach of dividing the coefficients into small square blocks and assigning an uniform quantizer q_k with dead-zone [4] to each block k, which means that if a coefficient α is located in block k the integer value $\operatorname{sgn}(\alpha)\lfloor \frac{\alpha}{q_k} \rfloor$ is encoded. The full details of the encoder will be given in Section 6. It remains to be decided what quantizer q_k to assign to each block k.

Following again Rate-Distortion Optimization [17], a widely adopted strategy is to fix a component-wise distortion metric D on the coefficients to be encoded, for example squared difference, and optimize over $\mathbf{q} = (q_1, \dots, q_k, \dots)$ the objective

$$\min_{\mathbf{q}} \sum_i D(\alpha_i, \tilde{\alpha}_{i,\mathbf{q}}) + \lambda_{\text{quant}} \operatorname{bits}(\tilde{\alpha}_{i,\mathbf{q}}) \qquad (5.1)$$

where $\tilde{\alpha}_{i,\mathbf{q}}$ is the quantized value of α_i under the choice of quantizers $\mathbf{q}$, and λ_{quant} is again a Lagrangian multiplier that trades off distortion for bitrate. Since each block can be optimized separately, the running time is linear in the number of blocks and quantizer choices.

One common choice for the distortion metric D is be the squared difference $D(x, y) = (x - y)^2$; if $\boldsymbol{\alpha} = W^T u$ is the vector of coefficients, the total distortion is equal to $\|\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}}_{\mathbf{q}}\|_2^2$; by orthogonality of W this is equal to $\|u - \tilde{u}_{\mathbf{q}}\|_2^2$ where $\tilde{u}_{\mathbf{q}} = W\tilde{\boldsymbol{\alpha}}_{\mathbf{q}}$, hence equal to the squared distortion of the field. By setting a strict bound on the average distortion (for example less than quarter-pixel precision) the quantized field can be made close enough to the real-valued field.

It is easy to see however that this can be very wasteful, as not all the motion vectors require the same precision. For example, in smooth areas of the image an imprecise motion vector should not induce a large error in the residual, while around sharp edges the vectors must be as precise as possible. This suggests that the precision required for the vectors should be related to the image gradient. We now formalize this intuition.

The ideal distortion we would like to optimize when quantizing a motion field is the *warping error* $\|I(u) - I(\tilde{u}_{\mathbf{q}})\|$ for some norm $\|\cdot\|$. This is not possible in the above framework because such distortion metric is non-separable as a function of the transformed coefficients. For this reason we derive a coefficient-wise *surrogate distortion metric* that approximates the warping error.

First, following again the approach in Section 4 we *linearize* the warping error around u, obtaining $\|\nabla I[u](u - \tilde{u}_{\mathbf{q}})\|$. As the quantization error is expected to be small, the linearization is a good approximation. Exploiting linearity, we can now rewrite it as $\|\nabla I[u]W(\boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}}_{\mathbf{q}})\| = \|\nabla I[u]W\tilde{e}\|$, where $\tilde{e} = \boldsymbol{\alpha} - \tilde{\boldsymbol{\alpha}}_{\mathbf{q}}$ is the quantization error. The argument of the norm is now linear in $\tilde{\boldsymbol{\alpha}}_{\mathbf{q}}$, but W introduces high-order dependencies between coefficients, hence this function cannot be used as a coefficient-wise distortion metric yet.

Let us now assume that the distortion $\|\cdot\|$ is L_2^2. If we can find a diagonal matrix $\Sigma = \operatorname{diag}(\sigma_1, \dots, \sigma_{2n})$ such that $\|\Sigma\tilde{e}\|_2$ approximates $\|\nabla I[u]W\tilde{e}\|_2$, then we can use as distortion metric $D_\Sigma(\alpha_i, \tilde{\alpha}_i) = \sigma_i^2(\alpha_i - \tilde{\alpha}_i)^2$ in (5.1) and obtain an approximation to the squared linearized warping error.

The squared linearized warping error can then be written as $\tilde{e}^T W^T \nabla I[u]^2 W\tilde{e}$. We choose as Σ the square root of the diagonal of $W^T \nabla I[u]^2 W$; if the contribution of the off-diagonal elements is small, meaning that $W^T \nabla I[u]^2 W$ is close to diagonal, the diagonal is a good approximation. We give an intuitive explanation of why this is the case: most vectors of the wavelet basis are localized in space, with most of their energy concentrated in a small number of pixels; hence in the matrix a large part of the energy is concentrated around the diagonal.

To compute Σ, note that $\sigma_i = \|\nabla I[u]We_i\|_2$, where e_i is the vector with 1 in i-th position and zeros elsewhere; in other words, σ_i is the norm of the i-th column of $\nabla I[u]W$. Since W is a multi-level wavelet transform, we do not have the matrix in explicit form, but an algorithm that computes the linear operator instead. However it is easy to see that if the wavelet has constant support, and the number of levels ℓ is constant, the columns have constant support. Furthermore, it is sufficient to compute the columns of W only on a $2^\ell \times 2^\ell$ square of the image domain, and the others can be obtained by translation. Hence we can compute them symbolically by modifying the inverse Fast Wavelet Transform algorithm [13], and compute the norms explicitly. The total time complexity is linear.

6. Implementation details

For our experiments we chose the family of least-asymmetric (Symlet) wavelets [5], specifically the Sym5 wavelet, which gave the best results. The Symlet wavelets are orthogonal, compactly supported, and almost symmetric; moreover, as most wavelets used in signal processing, they are continuous, hence any finite approximation of a signal by Symlet wavelets is continuous, regardless of the number of coefficients used. On the other hand, the Haar wavelet is discontinuous, thus it produces significant discontinuities in sparse approximations of continuous signals.

An example is shown in Figure 1: the Haar basis exhibits the same artifacts as block-based methods, while field and prediction obtained with Sym5 are smooth.

We use a decomposition with 5 detail levels (plus the approximation level). To choose the weights β_b in (4.1) the $\log n_b$ term suggests to use increments of 2 per level, because at each level the number of coefficients increase by a factor of 4; hence for the 6 levels we used $(2, 4, 6, 8, \infty, \infty)$ in all the experiments. We give infinite weight to the last two levels both to control the sparsity, and also because we estimate the field only on the luma component and use the same field on the chroma components; constraining the field to be locally smooth reduces the risk of overfitting to the luminance.

Field and residual encoder To evaluate experimentally the effectiveness of the method in a video compression setting we implemented a complete video encoder. This requires implementing an encoder for the residual image; we use

again a wavelet transform on the residual and then the same quantizer/entropy coder to encode the coefficients of both the field and the residual.

As quantizer/entropy coder we use a simplified version of Dirac's residual coder: the coefficients are split into blocks (we use 16×16 blocks) and each block is assigned an uniform quantizer q. When compressing the residual, each block of coefficients to be encoded can be taken either from the wavelet decomposition of the residual, or from the wavelet decomposition of the original frame; this is done to account for areas of the image where the residual is less compressible than the image itself, for example where the image is occluded in the previous frame. The entropy coder is the same context-based binary arithmetic coder used by Dirac, but we use a smaller number of contexts for predicting zeros and signs based on the neighboring coefficients already encoded. Unlike Dirac, for simplicity no coefficient prediction or zero prediction based on the parent subband are performed. Finally, the wavelet used in the decomposition is Sym5, instead of the biorthogonal wavelets used in Dirac.

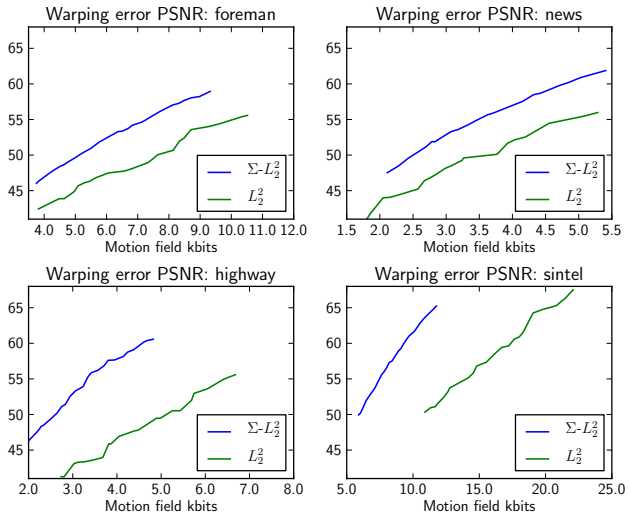

Figure 2. Rate-distortion curves of the field between the first two frames of the four video sequences used for evaluation. The y axis shows the PSNR between the reference warped with the continuous field and with the quantized field.

7. Experiments

Adaptive quantization We evaluated the quantization algorithm described in Section 5 on the motion field of the first pair of frames of four of the videos used in the experiments. The curves in Figure 2 show the PSNR of the actual warping error, measured as $\mathrm{PSNR}(I_1(u), I_1(\tilde{u}_{\mathbf{q}}))$ at different bit sizes (obtained by varying λ_{quant}), using the standard L_2^2 distortion metric and the weighted distortion metric Σ-L_2^2 described above. By optimizing for Σ-L_2^2 instead of L_2^2 the field can be encoded in roughly half the bits at the same quality . Furthermore, the rate-distortion curve obtained with the weighted distortion is significantly closer to linear, showing that the surrogate distortion is highly correlated with the actual warping error.

Experiments on the Middlebury dataset As an early synthetic benchmark, we used the grayscale image pairs of the Middlebury dataset [1] and measured the quality of the prediction against the compressed field size in bits, comparing with Horn-Schunck, as reported in Figure 3. The curves are obtained by finding the best quality/size trade-offs by a grid search on λ and λ_{quant}. L1Log and L2Log refer to our algorithm, with respectively L_1 and L_2^2 data terms, while HS is the classic Horn-Schunck; WQ variants use adaptive field quantization. On all pairs, our algorithm performs better than HS, obtaining fields that are as small as one half or one third than those obtained with HS, with the same prediction quality. Also, on almost all pairs, adaptive quantization significantly reduces the encoded field bitrate. Surprisingly, L1Log performs better than L2Log despite the metric is PSNR. We believe that this is caused by the robustness of the L_1 norm, that makes the optimization easier.

Video compression evaluation We compared our implementation against Dirac, a state-of-the-art wavelet video codec; its simplicity allowed us to implement a very similar encoder, so we can compare the contribution of the motion compensation component. Since our model can only handle pairs of frames, we configured Dirac to use a single reference frame; we will return again on the issue of multiple reference frames in Section 8. As in the Dirac encoder, no explicit rate control is performed; instead λ for the motion estimation and the λ_{quant} for field and residual quantization are fixed in advance and remain constant across the frames. For each sequence these parameters are searched on a grid to find the best quality at a given rate. The first frame of the sequence is encoded with no prediction, then each subsequent frame is predicted from the previous decoded frame. Both PSNR and bitrate are averaged over all the frames. We only compare the PSNR of the luma component, as it takes most of the bitrate; the chroma PSNR obtained is almost always greater than that of the luma in all the sequences.

Test sequences We performed our experiments on eight color (YUV420) sequences from a database of standard video compression test sequences [20]. They combine a variety of (camera and scene) motion types, and have different frame resolutions, 352×288 for foreman, news, highway, bus, flower, and football, 832×352 for sintel, and 704×576 for soccer.

Results Figure 5 compares the rate-distortion curves obtained with our implementations and Dirac on the first 25 frames of each test sequence. Despite the simplified residual encoder and the high level of tuning of the Dirac encoder,

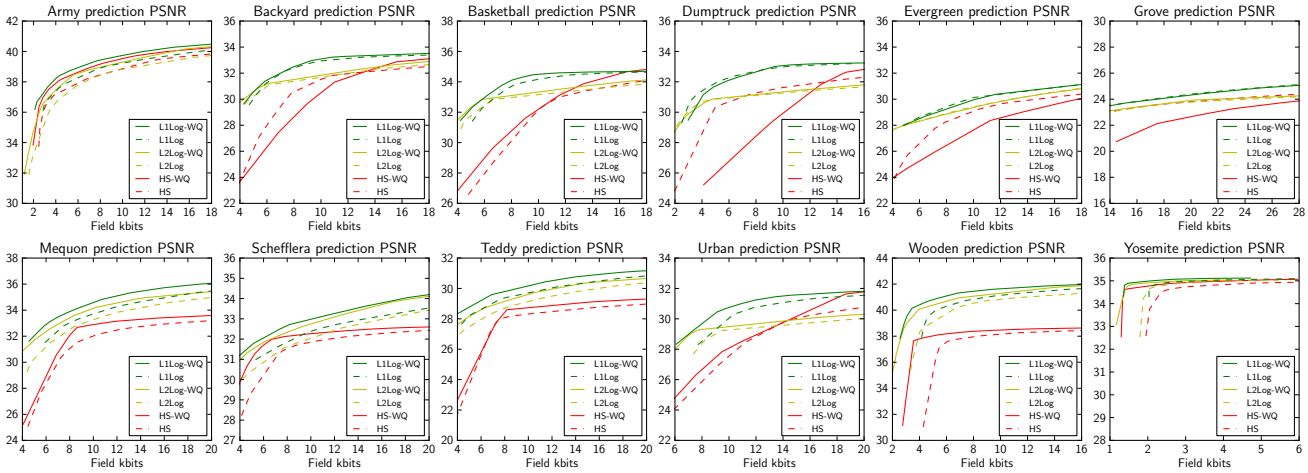

Figure 3. Results on the frame pairs of the Middlebury dataset. The x axis measures the compressed field size in kbits, the y axis is the PSNR between I_0 and $I_1(\tilde{u})$, i.e. the reference frame warped with the quantized field.

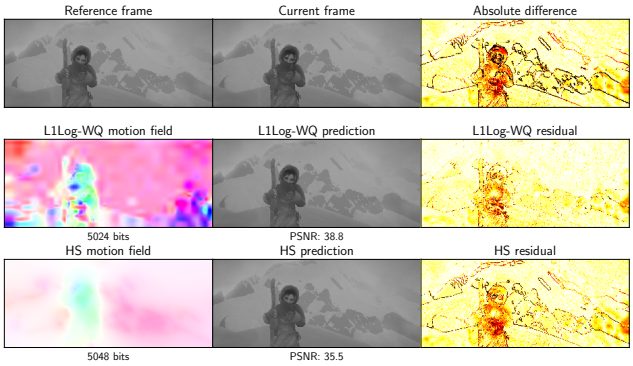

Figure 4. Fields obtained with our method (L1Log-WQ) and Horn-Schunck (HS) on the first two frames of sintel.

both L1Log-WQ and HS outperform it on all the sequences except bus at lower bitrates.

Compared to Dirac at the same PSNR, L1Log-WQ improves average bitrate by 23% on foreman, 38% on news, 47% on highway, 49% on sintel, 13% on flower, 39% on soccer, 5% on football, and 1% larger on bus.

L1Log-WQ also improves bitrate against HS on all the sequences, significantly on some: 11% on foreman, 7% on news, 8% on highway, 18% on sintel, 6% on bus, 5% on flower, 5% on soccer, 3% on football.

8. Discussion and Future Work

The experiments in Section 7 show that our method produces promising compression results in the single reference frame setting. It should be noted that the decoding complexity is only marginally higher than block-based motion compensation, as just a wavelet transform and a pixel-level warping are needed. Efficient decodability is a desirable property in many video compression applications, such as broadcasting, where the decoder must be implemented on low-power devices. With respect to encoding instead, our unoptimized implementation takes approximately 8 seconds for a 352×288 frame on an Intel Core 2. However we believe that a tuned GPU-based implementation could run in real-time. Whether our objective function can be efficiently optimized is an interesting open question.

One of the limitations of our current model is that it does not handle occlusions explicitly; instead it always tries to match each pixel of the current frame with some pixel of the reference frame. This may be a source of inefficiency in sequences with large occlusions, because the coefficients of the motion vectors in occluded areas are encoded anyway. Another limitation is the single reference frame constraint: multiple reference frames greatly improve the efficiency of video encoders.

We believe that it is possible to extend our model to consider both issues. Occlusions and lighting changes can be handled by adding a multiplicative per-pixel term l (when the term is 0, the pixel is considered occluded) and k references can be supported with a field component t ranging in the temporal dimension of the 3D image $I_{1,\dots,k}$. Thus the image prediction model becomes $I_{1,\dots,k}(u,t,l)$. Note that when both t and l are identically 1 the model reduces to the basic model $I_1(u)$ presented here. The terms t and l can be again transformed in a wavelet basis and quantized/encoded. The fields u, t and l should be optimized jointly, to find a good trade-off between prediction accuracy and their compressibility. However it not trivial to extend the optimization algorithm presented in Section 4 to this model. Another interesting question is how to generalize the algorithm to non-orthogonal bases, such as the biorthogonal wavelets widely used in image compression.

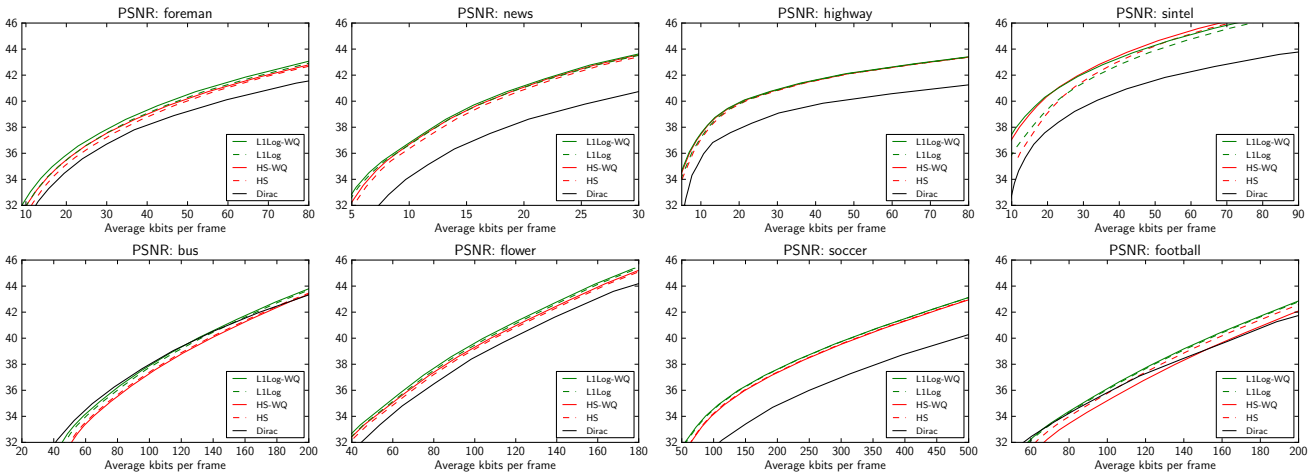

Figure 5. Rate-distortion curves on the test sequences, showing the average PSNR against the average bitrate per frame.

Acknowledgements

The authors would like to thank Antonio Criminisi, Andrew Fitzgibbon, Tom Minka, Sebastian Nowozin, and Christopher Zach for their invaluable suggestions and comments on this work.

References

[1] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1), 2011.

[2] E. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ^1 minimization. *Journal of Fourier Analysis and Applications*, 14, 2008.

[3] M. Chan, Y. Yu, and A. Constantinides. Variable size block matching motion compensation with applications to video coding. *Communications, Speech and Vision, IEEE*, 137(4), 1990.

[4] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley and Sons, Inc., 1991.

[5] I. Daubechies. Orthonormal bases of compactly supported wavelets ii: variations on a theme. *SIAM J. Math. Anal.*, 24(2), 1993.

[6] Dirac codec. http://diracvideo.org/download/specification/dirac-spec-latest.pdf.

[7] P. Elias. Universal codeword sets and representations of the integers. *IEEE Trans. on Information Theory*, 21(2), 1975.

[8] S.-C. Han and C. Podilchuk. Video compression with dense motion fields. *IEEE Trans. on Image Processing*, 10(11), 2001.

[9] B. K. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1–3), 1981.

[10] C.-L. Huang and C.-Y. Hsu. A new motion compensation method for image sequence coding using hierarchical grid interpolation. *IEEE Trans. on Circuits and Systems for Video Technology*, 4(1), 1994.

[11] J. Jain and A. Jain. Displacement measurement and its application in interframe image coding. *IEEE Trans. on Communications*, 29(12), 1981.

[12] S. Lin, Y. Shi, and Y.-Q. Zhang. An optical flow based motion compensation algorithm for very low bit-rate video coding. In *IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1997.

[13] S. Mallat. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 2008.

[14] D. Marpe, H. Schwarz, and T. Wiegand. Context-based adaptive binary arithmetic coding in the h.264/avc video compression standard. *IEEE Trans. on Circuits and Systems for Video Technology*, 13(7), 2003.

[15] P. Moulin, R. Krishnamurthy, and J. W. Woods. Multiscale modeling and estimation of motion fields for video coding. *IEEE Trans. on Image Processing*, 6(12), 1997.

[16] M. T. Orchard and G. J. Sullivan. Overlapped block motion compensation: an estimation-theoretic approach. *IEEE Trans. on Image Processing*, 3(5), 1994.

[17] G. Sullivan and T. Wiegand. Rate-distortion optimization for video compression. *Signal Processing Magazine, IEEE*, 15(6), 1998.

[18] H. Watanabe and S. Singhal. Windowed motion compensation. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 1991.

[19] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the h.264/avc video coding standard. *IEEE Trans. on Circuits and Systems for Video Technology*, 13(7), 2003.

[20] Xiph.org video test media. http://media.xiph.org/video/derf/, 2012.

[21] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L^1 optical flow. In *DAGM-Symposium*, 2007.