Wavelet-based Joint Estimation and Encoding of Depth-Image-based Representations for Free-Viewpoint Rendering

Matthieu Maitre, *Student Member, IEEE,* Yoshihisa Shinagawa, *Member, IEEE,* and Minh N. Do, *Member, IEEE*

***Abstract*— We propose a wavelet-based codec for the static Depth-Image-Based Representation (DIBR), which allows viewers to freely choose the viewpoint. The proposed codec jointly estimates and encodes the unknown depth map from multiple views using a novel Rate-Distortion (RD) optimization scheme. The rate constraint reduces the ambiguity of depth estimation by favoring piecewise-smooth depth maps. The optimization is efficiently solved by a novel dynamic programming along trees of integer wavelet coefficients. The codec encodes the image and the depth map jointly to decrease their redundancy and to provide a RD-optimized bitrate allocation between the two. The codec also offers scalability both in resolution and in quality. Experiments on real data show the effectiveness of the proposed codec.**

***Index Terms*— Free-viewpoint rendering, image-based rendering, 3D-TV, Depth-Image-Based Representation (DIBR), depth estimation, joint coding, scalable coding, rate-distortion optimization**

I. Introduction

FREE-VIEWPOINT Three-Dimensional Television (3D-TV) aims at providing an enhanced viewing experience not only by letting viewers perceive the third spatial dimension via stereoscopy but also by allowing them to move inside the 3D video and freely choose the viewing location they prefer [1]. The free-viewpoint approach is also useful for multi-user autostereoscopic 3D displays [2], which have to generate a large number of viewpoints.

The fundamental problem posed by 3D-TV lies in the massive amount of data required to represent the set of all possible views or, equivalently, the set of all light rays in the scene. This set of light rays, called the *plenoptic function* [3], lies in general in a seven-dimensional space. Each light ray travels along a line, which is described by a point (three dimensions), an angular orientation (two dimensions) and a time instant (one dimension). The last dimension describes the spectrum, or color, of the light rays. By comparison, 2D videos only lie in a four-dimensional space made of two angles, time, and color. Therefore, 3D-TV requires the design of a novel video chain [1].

M. Maitre is with the Department of Electrical and Computer Engineering, and the Beckman Institute, University of Illinois at Urbana-Champaign, Urbana IL 61801 (email: maitre@uiuc.edu).

Y. Shinagawa is with the Department of Electrical and Computer Engineering, and the Beckman Institute, University of Illinois at Urbana-Champaign, Urbana IL 61801 (email: sinagawa@uiuc.edu).

M. N. Do is with the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Beckman Institute, University of Illinois at Urbana-Champaign, Urbana IL 61801 (email: minhdo@uiuc.edu).

This work was supported by the National Science Foundation under Grant ITR-0312432.

A large number of methods have been proposed to record and encode the plenoptic function [4]. They widely differ in the amount of 3D geometry used to encode the data, which ranges from no geometry at all (e.g. light field) to an extremely accurate geometry (e.g. texture mapping). On the one hand, relying on the geometry has the advantage of requiring fewer cameras to record the plenoptic function and allowing the reduction of redundancies between the recorded views [5, 6]. On the other hand, using the geometry has the drawback of limiting the realism of the synthesized views and requiring a difficult estimation of the 3D geometry.

An efficient trade-off on the 3D geometry, called the *Depth-Image-Based Representation* (DIBR), consists in approximating the plenoptic function using pairs of images and depth maps [7]. Now part of the MPEG-4 standard [8, 9], this representation allows arbitrary views to be rendered in the vicinity of these pairs. Since depth maps tend to have lower entropies than images, the DIBR leads to compact bitstreams. Moreover, realistic images can be efficiently synthesized from the DIBR using Image-Based Rendering (IBR) and depth maps do not need to be estimated extremely accurately, as long as the viewpoint does not change too much.

Encoding the DIBR presents two difficulties. First, the depth maps are unknown. Therefore, not only do they have to be encoded, but they also have to be estimated. Second, the relation between the depth maps and the distortion of the plenoptic function is highly non-linear, which makes the Rate-Distortion (RD) optimization difficult. In particular, finding an optimal bitrate allocation between images and depth maps is non trivial.

A number of methods have avoided these issues by excluding depth maps from the RD problem. For instance, in [6, 10] depth maps are obtained using block-based depth estimation, essentially a motion estimation, and encoded in a *lossless* fashion. As an alternative to blocks, depth can also be estimated using meshes [11, 12] or pixel-wise regularization [7]. However, in such methods the image encoder and the depth encoder operate at different RD slopes, which penalizes the overall codec efficiency and makes it difficult to optimally allocate the bitrate [13].

A more principled approach consists in linearizing the RD problem [14, 15] using Taylor series expansions and statistical analysis. It has the advantage of leading to closed-form expressions and allowing a theoretical analysis of the problem. However, linearization is only valid for small depth approximations.

Another way of handling the non-linearity is to assume that depth maps take a finite number of discrete values. Under some constraints on the dependencies between depth values, globally optimal solutions can be found using Dynamic Programming (DP) [16]. For instance, optimal solutions exist when depth maps are encoded using Differential Pulse Code Modulation (DPCM) [17] or quadtrees [18]. This approach does not require any groundtruth: the estimation and encoding of the depth maps are carried out jointly. It also takes advantage of the bitrate constraint to favor smooth depth maps, much like ad-hoc smoothness terms do in computer vision [19], which reduces the ambiguity of the estimation.

There is a close relation between depth maps and 2D motion fields: depth maps define 3D surfaces, whose projection onto image planes gives rise to motion fields. Therefore, the techniques designed to solve the RD problem of classical 2D video coding [20] can usually also be applied to DIBR. Among these techniques, those described in [21, 22] are related to the proposed wavelet-based approach. In these codecs, images are split into blocks of variable sizes using quadtrees, and the motion vectors are DPCM coded. They achieve global optimality using DP. However, besides being not scalable, their complexity is exponential in the number of block sizes, which limits the range of block sizes they can handle.

In this paper, we propose a new wavelet-based DIBR codec which performs a RD-optimized encoding of multiple views. It differs from classical wavelet-based codecs in that part of the data to be transformed (i.e. the depth map) is *unknown*. Here, as shown in Figure 1, both the depth estimation, the depth encoding and the image encoding are performed jointly. Although the problem is non-linear, we present a codec able to efficiently find optimal solutions without resorting to linearization. We show that when the depth maps are represented using special integer wavelets, their joint estimation and coding via RD-optimization can be efficiently solved using DP along the tree of wavelet coefficients. The DP we introduce in this paper differ from the existing one of quadtrees [18, 23], as discussed in Section III-D. The RD-optimization of the integer wavelets favors piecewise-smooth depth maps, which reduces the estimation ambiguity and leads to compact representations of the data. The joint encoding of the images and depth maps provides a RD-optimized bitrate allocation. Furthermore, using the fact that depth discontinuities usually happen at image edges, it reduces the redundancies between depth maps and images by coding the two wavelet significance maps only once. The complexity of the proposed codec is only linear in the number of wavelet decomposition levels, due to the special tree structure we introduce.

In addition, the proposed codec offers scalability both in resolution, using wavelets, and in quality, using quality layers. The former allows servers to efficiently stream data to display devices with inhomogeneous display resolutions and inside online virtual 3D worlds, where the DIBR may actually only cover a small portion of the display due to its distance to the viewpoint. The latter lets servers efficiently stream data over networks with inhomogeneous capabilities. In both cases, the RD point is chosen on the fly at the server by truncating the bitstream [24].

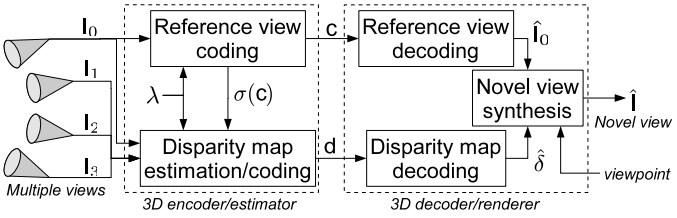


Fig. 1. Overview of the proposed codec. The encoder jointly estimates and encodes a DIBR at the RD slope λ from the set of views $\{\mathsf{I}_v\}$. The image and the disparity map are represented by the vectors of wavelet coefficients c and d respectively, whose joint entropy is reduced by sharing the significance map $\sigma(\mathsf{c})$. At the decoder, an image $\hat{\mathsf{I}}_0$ and a disparity map $\hat{\delta}$ are reconstructed, which allow novel views $\hat{\mathsf{I}}$ to be synthesized from arbitrary viewpoints.

The remainder of the article is organized as follows. Section II describes the proposed codec and the RD problem at hand, while Section III details the RD optimization of the DIBR. Finally, Section IV presents our experimental results. Some preliminary results appeared in [25].

II. Proposed Codec

First, we define the RD problem that shall be solved by the proposed codec. As illustrated in Figure 1, the encoder takes a set of synchronized views as input and represents them using the DIBR. The decoder receives the DIBR and synthesizes novel views at 3D locations chosen by the viewers.

The DIBR consists in a subset of the views, called reference views, along with unknown depth maps. In the following, we limit our study to the case of static grayscale views. In this case, the DIBR provides an approximation to five-dimensional plenoptic functions with three spatial dimensions and two angular dimensions. Since the DIBR only offers a local approximation of the plenoptic function, the viewers are free to choose arbitrary viewpoints, but only inside a Neighborhood Of Views (NOV). A natural choice for the shape of the NOV is to take the union of a set of hypervolumes made of 3D spheres in space and 2D discs in angle, with one hypervolume associated to each pair of image and depth map. Since the approximation does not usually degrade abruptly when the distance increases, the decoder could actually enforce a "soft" NOV boundary by discouraging the viewer to choose a viewpoint outside of the NOV without forbidding it.

The distortion introduced by the codec is measured using the Mean-Square Error (MSE) between the recorded views and the views rendered from the DIBR. Denoting the v^{th} recorded view and its rendered counterpart by respectively the column vectors I_v and $\hat{\mathsf{I}}_v$ obtained by stacking all the pixels together, the distortion can be written as

$$D(\{\mathsf{I}_v\},\{\hat{\mathsf{I}}_v\}) \triangleq \frac{1}{N_m N_n N_v} \sum_{v=0}^{N_v-1} \left\| \mathsf{I}_v - \hat{\mathsf{I}}_v \right\|_2^2 \qquad (1)$$

where $\|.\|_2$ denotes the 2-norm, N_m and N_n are respectively the number of rows and columns in the views and N_v is the number of views. We denote by $N \triangleq N_m N_n N_v$ the total number of pixels. The distortion takes into account errors on both the reference views included in the DIBR and the novel views rendered from the DIBR.

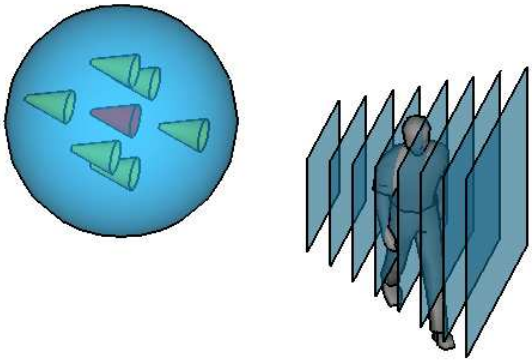
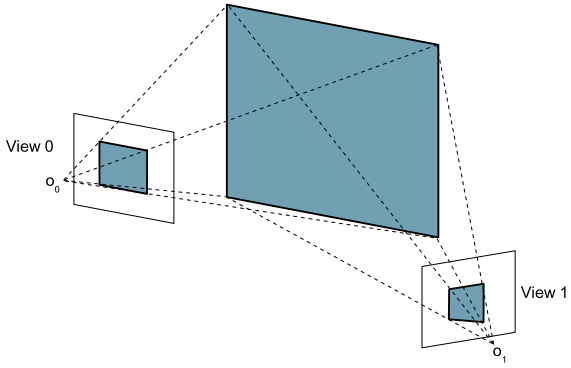
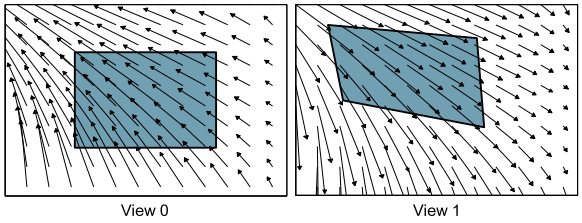


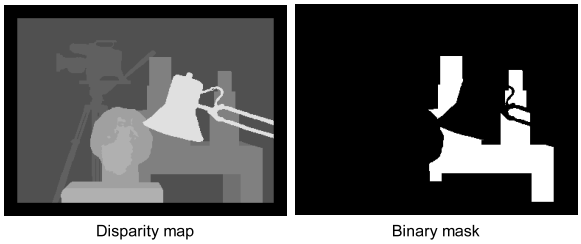
Fig. 2. The spatial extent of a NOV (sphere) with one pair of image and depth map, along with seven views (cones). The central dark cone designates the reference view. The planes represent iso-depth surfaces (3D model courtesy of Google).



(a) A reference view $s = 0$ along with an arbitrary view $s = 1$ and an iso-depth plane.



(b) The two views and the associated motion fields.



(c) A disparity map $\hat{\delta}$ and the binary mask $\mathsf{m}(\hat{\delta}, d)$ associated with an iso-depth planes d.

Fig. 3. The projection of an iso-depth plane onto two views gives rise to a motion field between the two which is a 2D homography. Only the pixels with non-zero values in the binary mask $\mathsf{m}(\hat{\delta}, d)$ are kept after motion compensation.

The decoder renders novel views using the nearest pair of image and depth map. The total distortion is then the sum of the distortions associated with each pair of image and depth map. Likewise, the pairs of images and depth maps are encoded independently of one another, so that the total bitrate is also the sum of the bitrates associated with each pair.

As a consequence, the RD problem can be solved for each pair of image and depth map independently. Rendering and encoding multiple pairs jointly might lead to increased RD performances. However, this would introduce complex data dependencies which cannot be trivially handled by the proposed method. Moreover, this would reduce the ability of the decoder to access views randomly [4]. Without loss of generality, the remainder of this article only considers the case where a unique pair is encoded and the reference view is indexed by $v = 0$.

The quantized depth map takes a finite number of discrete values, which define a set of iso-depth planes, as shown in Figure 2. Each plane induces a special motion field between the reference view and an arbitrary view which is a *homography* [26], as shown in Figure 3. This class of motion fields has the property of transforming quadrilaterals into quadrilaterals and includes affine transforms as a special case. In the particular case of rectified views [26], the motion vectors are parallel to the baseline of the pair of views.

In this framework, the depth estimation is formulated in terms of disparities, which are inversely proportional to depths. Disparities are better suited to the geometry of the problem at hand. They take into account the decreasing accuracy of the depth estimation as depth increases and they are equal to motion vectors in the case of rectified views.

Both the reference view and the disparity map are encoded in a *lossy* manner. Let us denote the encoded reference view by the vector $\hat{\mathsf{I}}_0$ and the jointly estimated and encoded disparity map by the vector $\hat{\delta}$. The view I_v is approximated by forward motion compensation of the reference view $\hat{\mathsf{I}}_0$ using the estimated disparity map $\hat{\delta}$, an operation denoted by $\mathcal{M}_v^f(\hat{\mathsf{I}}_0; \hat{\delta})$ where the f stands for 'forward'.

The forward motion compensation applies the set of iso-depth homographies to the reference view $\hat{\mathsf{I}}_0$ using an accumulation buffer and multiple texture-mapping operations, which benefits from hardware acceleration [27]. The accumulation buffer consists in a memory buffer, which is initially empty and progressively filled by the intensity values of texture-mapped views. For each discrete disparity value d, the following three steps are taken:

1) a binary mask $\mathsf{m}(\hat{\delta}, d)$ is defined, as shown in Figure 3, which takes value one at pixels with disparity value d,
2) the homography associated with the disparity value d is applied to both the reference view $\hat{\mathsf{I}}_0$ and the mask $\mathsf{m}(\hat{\delta}, d)$ using texture mapping,
3) the values of the pixels in the accumulation buffer for which the motion-compensated mask is one are replaced by those of the motion-compensated view.

The disparities are processed in decreasing depth order so that in step 3 the textures from closer iso-depth planes replace those from further iso-depth planes, therefore correctly enforcing the occlusion relations between iso-depth planes. The issues of resampling and hole filling are solved using bilinear interpolation and texture propagation by Poisson equation [28].

We shall encode both the image $\hat{\mathsf{I}}_0$ and the disparity map $\hat{\delta}$ in the wavelet domain. Let c and d be the column vectors of their wavelet coefficients, respectively. The wavelet synthesis operators relate these vectors by

$$\hat{\mathsf{I}}_0 \triangleq \mathbf{T}\mathsf{c} \quad \text{and} \quad \hat{\delta} \triangleq \mathcal{T}(\mathsf{d}), \tag{2}$$

where the matrix $\mathbf{T}$ represents the linear wavelet transform for the image and the function $\mathcal{T}$ represents the integer-to-integer wavelet transform for the discrete-valued disparity map.

We define two significance maps, $\sigma(\mathsf{c})$ and $\sigma(\mathsf{d})$, which are binary vectors with value one in the presence of non-null wavelet coefficients and zero otherwise. These maps are not directional, i.e. they are the same for all directional subbands at each scale of the 2D wavelet transform. In this way, we shall be able to compare $\sigma(\mathsf{c})$ and $\sigma(\mathsf{d})$ even when the wavelet operators $\mathbf{T}$ and $\mathcal{T}$ differ in their directional division of the frequency plane.

In natural images, discontinuities in the disparity map are usually associated with discontinuities in the image. When they are not, it is very difficult to estimate the disparity discontinuities from multiple views. Therefore, we can reduce the data redundancy of the DIBR by coding the image and the disparity significance maps jointly. This is done by coding only the significance map of the image $\sigma(\mathsf{c})$ and assuming the significant coefficients in $\sigma(\mathsf{d})$ to be a subset of those in $\sigma(\mathsf{c})$, that is,

$$\sigma_n(\mathsf{c}) = 0 \Rightarrow \sigma_n(\mathsf{d}) = 0, \ \forall n \in [0, N_m N_n - 1] \tag{3}$$

This joint encoding also reduces the complexity of the search for the optimal vector d^* by fixing a large number of its coefficients to zero.

The total rate $R(\mathsf{c}, \mathsf{d})$ is then given by the sum of the rates $R(\mathsf{c})$, and $R(\mathsf{d}|\sigma(\mathsf{c}))$ and the RD problem is

$$\begin{aligned} &\min_{\mathsf{c},\mathsf{d}} \frac{1}{N} \sum_{v=0}^{N_v-1} \left\| \mathsf{I}_v - \mathcal{M}_v^f(\mathbf{T}\mathsf{c}; \mathcal{T}(\mathsf{d})) \right\|_2^2 \\ &\text{such that } R(\mathsf{c}) + R(\mathsf{d}|\sigma(\mathsf{c})) \leq R_{max} \end{aligned} \tag{4}$$

where R_{max} is the maximum rate allowed. The constraint (3) appears implicitly in the rate constraint: $R(\mathsf{d}|\sigma(\mathsf{c}))$ takes the value $+\infty$ when this constraint is violated. Introducing the Lagrange multiplier λ [20], (4) can be written as

$$\min_{\mathsf{c},\mathsf{d}} \frac{1}{N} \sum_{v=0}^{N_v-1} \left\| \mathsf{I}_v - \mathcal{M}_v^f(\mathbf{T}\mathsf{c}; \mathcal{T}(\mathsf{d})) \right\|_2^2 + \lambda \left(R(\mathsf{c}) + R(\mathsf{d}|\sigma(\mathsf{c})) \right). \tag{5}$$

This equation has three goals. First, it encodes the reference view. Second, it estimates the disparity map, and therefore allows the rendering of arbitrary views. Finally, it encodes this disparity map. Solving this optimization shall be the topic of the next section.

III. Rate-Distortion Optimization

A. Overview

Since (5) is non linear, solving it is not a trivial operation. Therefore, we formulate several approximations to obtain a computationally efficient method.

First, we ignore the issues of occlusions and resampling. In this way, the motion-compensation operation becomes invertible and the optimization problem can be defined either on the rendered views or on the reference view. The latter option turns out to be much more practical because it decouples the encoded reference view from the motion compensation. Mathematically, this assumption is equivalent to

$$\left\| \mathsf{I}_v - \mathcal{M}_v^f(\hat{\mathsf{I}}_0; \hat{\delta}) \right\|_2^2 \approx \left\| \mathcal{M}_v^b(\mathsf{I}_v; \hat{\delta}) - \hat{\mathsf{I}}_0 \right\|_2^2 \tag{6}$$

where $\mathcal{M}_v^b(\mathsf{I}_v; \hat{\delta})$ denotes the backward-motion-compensated view I_v. Equation (5) then becomes

$$\min_{\mathsf{c},\mathsf{d}} \frac{1}{N} \sum_{v=0}^{N_v-1} \left\| \mathcal{M}_v^b(\mathsf{I}_v; \mathcal{T}(\mathsf{d})) - \mathbf{T}\mathsf{c} \right\|_2^2 + \lambda \left(R(\mathsf{c}) + R(\mathsf{d}|\mathsf{c}) \right). \tag{7}$$

The MSE term in (7) depends on the wavelet vectors c and d in very different ways: it is quadratic in c but not in d. Therefore, the minimization is solved using coordinate descent [29], first minimizing c and then d. The minimization of c ignores the dependency with d due to the shared significance map. This shall allow us to use classical wavelet coding techniques for c and dynamic programming for d.

The optimization is initialized at high bitrate where the MSE is virtually null, that is,

$$\mathbf{T}\mathsf{c} \approx \mathsf{I}_0 \text{ and } \mathcal{M}_v^b(\mathsf{I}_v; \mathcal{T}(\mathsf{d})) \approx \mathsf{I}_0. \tag{8}$$

In general, we would need to iterate the successive optimization process until convergence. Here, however, only one iteration is run to reduce the computational complexity and prevent erroneous disparities from introducing blur in the encoded reference view $\hat{\mathsf{I}}_0$.

In the remainder of this section, we first describe the optimization of the reference view in Section III-B. We then detail the optimization of the disparity map, beginning with the simpler case of one-dimensional views in Section III-C, which we extend to two-dimensional views in Section III-E. Finally, we present how a quality scalable bitstream can be obtained in Section III-G.

B. Reference View

We start with the optimization of the wavelet coefficients c of the reference view. Fixing d and using the high-bitrate assumption (8), the optimization problem (7) becomes

$$\min_{\mathsf{c}} \frac{1}{N_m N_n} \left\| \mathsf{I}_0 - \mathbf{T}\mathsf{c} \right\|_2^2 + \lambda R(\mathsf{c}). \tag{9}$$

When the wavelet transform $\mathbf{T}$ is nearly orthonormal, like the 9/7 wavelet [30] for instance, this equation can be further simplified to

$$\min_{\mathsf{c}} \frac{1}{N_m N_n} \left\| \mathbf{T}^{-1}\mathsf{I}_0 - \mathsf{c} \right\|_2^2 + \lambda R(\mathsf{c}), \tag{10}$$

which is a standard problem in image compression and is readily solved by wavelet-based coders [24].

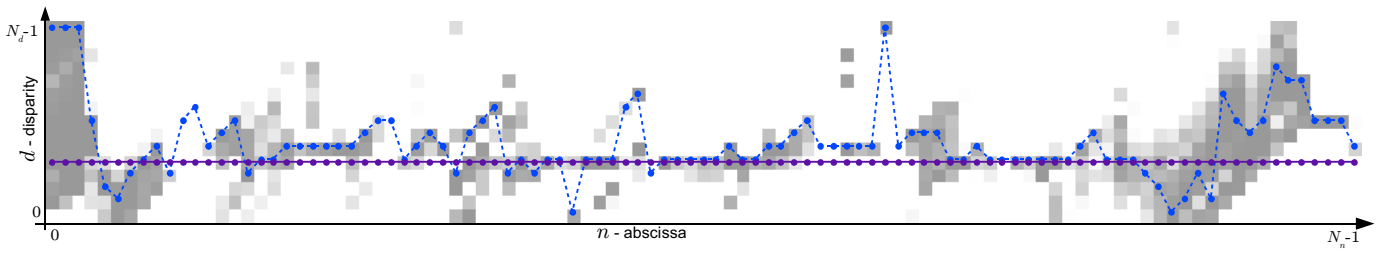


Fig. 4. An error matrix $\mathbf{E}$ from the Tsukuba image set [19] with two optimal paths overlaid, $\lambda = 0$ (dashed) and $\lambda = \infty$ (solid). Lighter shades of gray indicate larger squared intensity differences.

C. One-Dimensional Disparity Map

The next step is to find an optimal solution for the wavelet coefficients d of the disparity map. To begin with, we consider the special case of one-dimensional views ($N_m = 1$).

Fixing c, the optimization problem (7) becomes

$$\min_{\mathsf{d}} \frac{1}{N} \sum_{v=0}^{N_v-1} \sum_{n=0}^{N_n-1} \left(\mathcal{M}^b_{v,n}(\mathsf{I}_v; \hat{\delta}_n) - \hat{\mathsf{I}}_{0,n} \right)^2 + \lambda R(\mathsf{d}|\sigma(\mathsf{c})) \tag{11}$$

where $\hat{\mathsf{I}}_0 \triangleq \mathbf{T}\mathsf{c}$ and $\mathcal{M}^b_{v,n}(\mathsf{I}_v; d)$ denotes the intensity value of the pixel in I_v which would correspond to the pixel n in the reference view if the pixel n had the disparity value d. Unlike in the previous section, the MSE term is not a quadratic function of the wavelet coefficients, due to the non-linearity of motion compensation. Instead, we take advantage of the fact that the disparity map only takes a finite number of values.

The MSE term can be written in terms of an error matrix $\mathbf{E}$ in which the entry $\mathbf{E}_{d,n}$ gives the scaled square error that the pixel n of $\hat{\mathsf{I}}_0$ would be associated with if it had disparity d (see Figure 4). That is,

$$\mathbf{E}_{d,n} \triangleq \frac{1}{N} \sum_{v=0}^{N_v-1} \left(\mathcal{M}^b_{v,n}(\mathsf{I}_v; r) - \hat{\mathsf{I}}_{0,n} \right)^2. \tag{12}$$

This error matrix is also called "disparity space image" [19] and is independent of the disparity map $\hat{\delta}$. Computing this matrix has a complexity of $O(NN_d)$ where N_d is the number of disparity values.

We study the encoding of the disparity map using two transforms, namely the Sequential (S) transform [24] and a transform we call the Laplace (L) transform due to its resemblance to the Laplacian pyramid [31]. Both provide a compact representation of discrete and piecewise-constant disparity maps. Both also induce graphs of dependencies between their wavelet coefficients as trees, so that the problem can be efficiently solved using dynamic programming [32]. They differ in their redundancy, the former being non redundant. They also differ in the complexity of their optimization with regard to the number of disparity values N_d, the latter being of linear complexity and the former of quadratic complexity.

The analysis and synthesis operators of these two transforms are given in Table I, where $\lfloor x \rfloor$ denotes the largest integer less or equal to x. They relate the low-pass coefficients l and the high-pass coefficients h between the level $j-1$ with finer resolution and the level j with coarser resolution.

The wavelet vector d is made of all the high-pass coefficients h, along with the low-pass coefficients l of the coarsest level $j \triangleq L-1$. The low-pass coefficients l at the finest level $j \triangleq 0$ are equal to the disparities $\hat{\delta}$, that is, $l^{(0)}_n = \hat{\delta}_n$.

The probability of the wavelet coefficients is approximated as follows. The coefficients l at the coarsest level and the coefficients h are assumed to be jointly independent. The coefficients l follow a uniform distribution. The coefficients h are null with probability one if the corresponding image coefficients are insignificant and otherwise follow a discrete and truncated Laplace distribution with zero mean [24], that is,

$$p(h^{(j)}_n | \sigma^{(j)}_n(\mathsf{c})) = \begin{cases} \mathbf{1}_{h^{(j)}_n = 0} & \text{if } \sigma^{(j)}_n(\mathsf{c}) = 0, \\ \frac{1}{Z} e^{-\frac{|h^{(j)}_n|}{b}} \mathbf{1}_{|h^{(j)}_n| \leq N_d} & \text{otherwise,} \end{cases} \tag{17}$$

where b is a parameter to be estimated and Z is a normalizing constant. This probability distribution defines the entropy of the data [33], that we use as an approximation of the actual bitrate. The bitrate, in bits per pixel, is therefore

$$R(\mathsf{d}|\sigma(\mathsf{c})) = -\sum_{j=0}^{L-1} \sum_{n=0}^{N_h(j)-1} \log_2 \left(p(h^{(j)}_n | \sigma^{(j)}_n(\mathsf{c})) \right) + cst \tag{18}$$

where $\log_2$ denotes the logarithm to base 2, cst is a constant term independent of d and $N_h(j)$ is the number of high-pass coefficients at level j. We introduce the cost function

$$C(h^{(j)}_n) \triangleq \begin{cases} +\infty & \text{if } \sigma^{(j)}_n(\mathsf{c}) = 0 \text{ and } h^{(j)}_n \neq 0, \\ \mu |h^{(j)}_n| & \text{otherwise} \end{cases} \tag{19}$$

where $\mu \triangleq \lambda / (b \log 2)$ acts as a smoothness factor. Using this cost function, the equation of the bitrate (18) becomes

$$R(\mathsf{d}|\sigma(\mathsf{c})) = \frac{1}{\lambda} \sum_{j=0}^{L-1} \sum_{n=0}^{N_h(j)-1} C(h^{(j)}_n) + cst \tag{20}$$

and the optimization problem (11) can be written as

$$\min_{\mathsf{d}} \sum_{n=0}^{N_n-1} \mathbf{E}_{l^{(0)}_n, n} + \sum_{j=0}^{L-1} \sum_{n=0}^{N_h(j)-1} C(h^{(j)}_n). \tag{21}$$

D. Dynamic Programming

The optimization problem (21) is still a minimization over a space with large dimension. However, it can be solved recursively by a series of minimizations over small search spaces. The approach consists in using the commutativity

L-transform		S-transform	
analysis	synthesis	analysis	synthesis
$l_n^{(j)} = \left\lfloor \frac{l_{2n}^{(j-1)} + l_{2n+1}^{(j-1)}}{2} \right\rfloor$ $h_{2n}^{(j-1)} = l_{2n}^{(j-1)} - l_n^{(j)}$ $h_{2n+1}^{(j-1)} = l_{2n+1}^{(j-1)} - l_n^{(j)}$ (13)	$l_{2n}^{(j-1)} = l_n^{(j)} + h_{2n}^{(j-1)}$ $l_{2n+1}^{(j-1)} = l_n^{(j)} + h_{2n+1}^{(j-1)}$ (14)	$l_n^{(j)} = \left\lfloor \frac{l_{2n}^{(j-1)} + l_{2n+1}^{(j-1)}}{2} \right\rfloor$ $h_n^{(j)} = l_{2n}^{(j-1)} - l_{2n+1}^{(j-1)}$ (15)	$l_{2n}^{(j-1)} \overset{\mathcal{S}_0}{=} l_n^{(j)} - \left\lfloor \frac{h_n^{(j)}}{2} \right\rfloor + h_n^{(j)}$ $l_{2n+1}^{(j-1)} \overset{\mathcal{S}_1}{=} l_n^{(j)} - \left\lfloor \frac{h_n^{(j)}}{2} \right\rfloor$ (16)

TABLE I

ANALYSIS AND SYNTHESIS OPERATORS OF THE LAPLACE (L) AND SEQUENTIAL (S) TRANSFORMS (SEE TEXT FOR DETAILS).

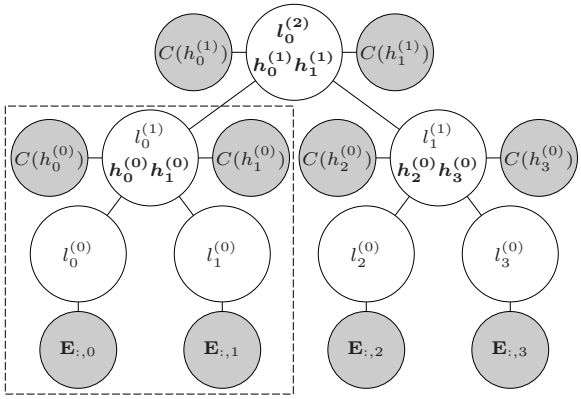


Fig. 5. Dependency graph of a three-level L transform. The coefficients in bold are those included in the wavelet vector d. Gray nodes represent the MSE and rate terms of the RD optimization. The dashed box highlights the two-level L transform associated with (22).

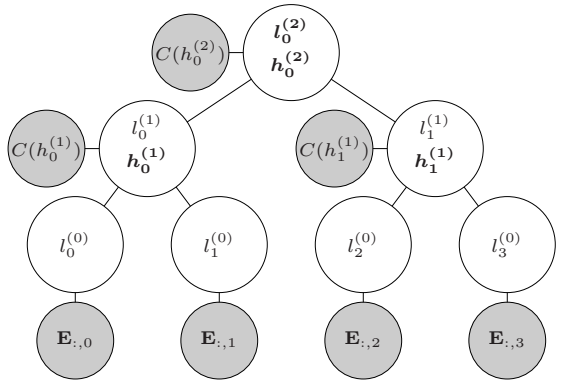


Fig. 6. Dependency graph of a three-level S transform. The coefficients in bold are those included in the wavelet vector d. Gray nodes represent the MSE and rate terms of the RD optimization.

of the sum and min operators to group the terms of the summation together based on the variables they depend on. This is possible due to the choice of wavelets, which do not introduce loops in the dependency graph of the group of terms, as shown in Figures 5 and 6. In these figures, the notation $\mathbf{E}_{:,n}$ denotes the column n of the error matrix $\mathbf{E}$. This column vector contains the errors of the different disparity values at the pixel location n.

Example 1: Let us consider a simple example to illustrate the algorithm. The optimization problem associated with a two-level L transform, emphasized by a dashed box in Figure 5, is given by

$$\min_{l_0^{(1)}, h_0^{(0)}, h_1^{(1)}} \Big(\mathbf{E}_{l_0^{(1)} + h_0^{(0)}, 0} + \mathbf{E}_{l_0^{(1)} + h_1^{(0)}, 1} + C(h_0^{(0)}) + C(h_1^{(0)}) \Big). \tag{22}$$

By grouping the terms of the summation together and commuting the min and sum operators, it can be rewritten as

$$\min_{l_0^{(1)}} \Big(\min_{h_0^{(0)}} \left(\mathbf{E}_{l_0^{(1)} + h_0^{(0)}, 0} + C(h_0^{(0)}) \right) + \min_{h_1^{(0)}} \left(\mathbf{E}_{l_0^{(1)} + h_1^{(0)}, 1} + C(h_1^{(0)}) \right) \Big) \tag{23}$$

which reduces the complexity from cubic to quadratic.

Next, we illustrate how to solve the inner minimizations. Let us consider the minimization over $h_0^{(0)}$ with a smoothness factor $\mu = 0.5$. We assume that the disparity values range from 0 to 5 and solve for the case $l_0^{(1)} = 2$. Let us assume that the first column vector of the error matrix $\mathbf{E}$ is

$l_0^{(0)}$	0	1	2	3	4	5
$\mathbf{E}_{:,0}^{\mathsf{T}}$	2	5	3	0.25	4	2

(24)

Stacking the values of the cost function $C(h_0^{(0)})$ for each $h_0^{(0)}$ (note that $h_0^{(0)} = l_0^{(0)} - l_0^{(1)} = l_0^{(0)} - 2$) into a cost vector C using (19) gives

$l_0^{(0)}$	0	1	2	3	4	5
$h_0^{(0)}$	-2	-1	0	1	2	3
C^{T}	1	0.5	0	0.5	1	1.5

(25)

The sum of these two vectors is

$l_0^{(0)}$	0	1	2	3	4	5
$\mathbf{E}_{:,0}^{\mathsf{T}} + \mathsf{C}^{\mathsf{T}}$	3	5.5	3	0.75	5	3.5

(26)

The minimum is therefore 0.75, which is reached at $l_0^{(0)} = 3$. By the definition of the synthesis operator (14), it follows that the optimal high-pass coefficient associated with $l_0^{(1)} = 2$ is $h_0^{(0)} = 1$. ∎

In the general case, the recursive minimization is defined using a pyramid of error matrices $\{\mathbf{E}^{(j)}, j \in [0, \ldots, L-1]\}$. The error matrix at the finest level $j = 0$ is defined by

$$\mathbf{E}^{(0)} \triangleq \mathbf{E}. \tag{27}$$

The error matrices of the L transform at coarser levels are given by

$$\begin{aligned}\mathbf{E}_{d,n}^{(j)} = &\min_{h_{2n}^{(j-1)}} \left(\mathbf{E}_{d+h_{2n}^{(j-1)},2n}^{(j-1)} + C(h_{2n}^{(j-1)})\right) \\ &+ \min_{h_{2n+1}^{(j-1)}} \left(\mathbf{E}_{d+h_{2n+1}^{(j-1)},2n+1}^{(j-1)} + C(h_{2n+1}^{(j-1)})\right).\end{aligned} \tag{28}$$

The error matrices of the S transform at coarser levels are given by

$$\mathbf{E}_{d,n}^{(j)} = \min_{h_n^{(j)}} \left(\mathbf{E}_{\mathcal{S}_0(d,h_n^{(j)}),2n}^{(j-1)} + \mathbf{E}_{\mathcal{S}_1(d,h_n^{(j)}),2n}^{(j-1)} + C(h_n^{(j)})\right) \tag{29}$$

where $\mathcal{S}_0(.)$ and $\mathcal{S}_1(.)$ denote the synthesis operators defined by (16).

Computing an error matrix $\mathbf{E}^{(j)}$ of the S transform has a complexity quadratic in the number of disparity values N_d: error values need to be computed for each disparity value d and each value of the high-pass coefficient $h_n^{(j)}$. On the other hand, an error matrix $\mathbf{E}^{(j)}$ of the L transform can be computed with only linear complexity, as was shown in [34] in the case of Markov random fields with linear smoothness function.

The optimization problem (21) becomes simply

$$\min_{l_n^{(L-1)}} \mathbf{E}_{l_n^{(L-1)},n}^{(L-1)} \tag{30}$$

for each low-pass coefficient $l_n^{(L-1)}$ at the coarsest level $L-1$.

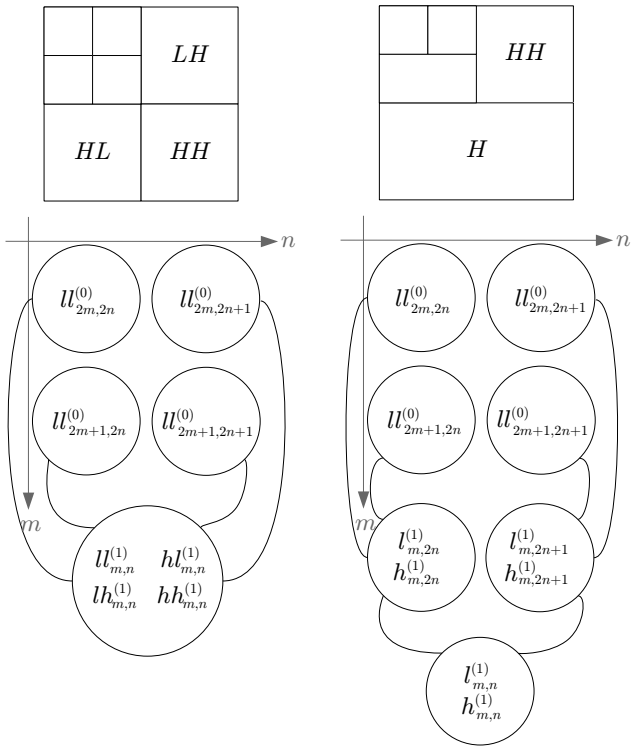
The pyramid of error matrices is associated with a pyramid of matrices of high-pass coefficients $\{\mathbf{H}^{(j)}, j \in [0,\ldots,L-1]\}$. At each level, they store the high-pass coefficients which achieve the minima in (28) or (29). Once the optimal low-pass coefficients $l_n^{(L-1)*}$ are known, the low-pass and high-pass coefficients at other levels are obtained by backtracking using the matrices $\mathbf{H}^{(j)}$ and the synthesis operators (16) or (14).

Therefore, the overall algorithm to solve (21) is the following:

1) the *initialization* creates the error matrix $\mathbf{E}^{(0)}$,
2) the *bottom-up pass* computes the matrices $\mathbf{E}^{(j)}$ and $\mathbf{H}^{(j)}$,
3) the *coarsest-level minimization* finds the optimal low-pass coefficients $l_n^{(L-1)*}$,
4) the *top-down pass* backtracks to compute the optimal low-pass and high-pass coefficients $l_n^{(j)*}$ and $h_n^{(j)*}$ at all levels.

At the end, both the globally optimal disparity map $\hat{\delta}^*$ and the globally optimal wavelet vector d^* are known. The initialization has a complexity of $O(NN_d)$, the bottom-up pass of $O(N_nN_d^2)$ in the case of the S transform and $O(N_nN_d)$ in the case of the L transform, the coarsest-level minimization of $O(N_nN_d)$ and the top-down pass of $O(N_n)$.

This algorithm shares similarities with quadtree-based motion estimation [23]. In quadtree estimation, small minimizations are solved at each node of the tree to find optimal motion vectors and decide whether to split or merge. In the proposed algorithm, small minimizations are also solved at each node, but they find optimal wavelet coefficients instead. A major difference between quadtrees and wavelets lies in the way the data is stored in the tree. Quadtrees store the data at their leaves using independent coefficients, while wavelets spread the data over the entire tree using differential coefficients. Therefore, wavelets offer resolution scalability while quadtrees do not. Another difference lies in the induced smoothness. Quadtrees enforce constant values inside blocks but no smoothness between blocks, while wavelets induce a smoothness between all pixels. Our experimental results shall show that the latter reduces spurious noise in the estimated disparity maps.



(a) Four-band division (b) Three-band division

Fig. 7. Two divisions of the frequency plane and the associated graphs of dependencies between the coefficients of the S transform.

E. Two-Dimensional Disparity Map

We now extend the optimization procedure to two-dimensional views. The error matrix $\mathbf{E}_{d,n}^{(0)}$ becomes an error tensor $\mathbf{E}_{d,m,n}^{(0)}$ with three dimensions: rows m, columns n and disparities d. It is defined as

$$\mathbf{E}_{d,m,n}^{(0)} \triangleq \frac{1}{N} \sum_{v=0}^{N_v-1} \left(\mathcal{M}_{v,m,n}^{b}(\mathsf{I}_v; d) - \hat{\mathsf{I}}_{0,m,n}\right)^2. \tag{31}$$

Its computation has a complexity of $O(NN_d)$, which remains linear in all the variables.

The two-dimensional extension of the L transform is also straightforward. Its synthesis operator (14) simply becomes

$$\begin{cases} l_{2m,2n}^{(j-1)} = l_{m,n}^{(j)} + h_{2m,2n}^{(j-1)} \\ l_{2m+1,2n}^{(j-1)} = l_{m,n}^{(j)} + h_{2m+1,2n}^{(j-1)} \\ l_{2m+1,2n+1}^{(j-1)} = l_{m,n}^{(j)} + h_{2m+1,2n+1}^{(j-1)} \\ l_{2m,2n+1}^{(j-1)} = l_{m,n}^{(j)} + h_{2m,2n+1}^{(j-1)} \end{cases} \tag{32}$$

The computational complexity at each node of the dependency tree remains $O(N_d)$, with a total complexity of $O(N_m N_n N_d)$ for the bottom-up pass.

The two-dimensional extension of the S transform is slightly more complex. We follow the classical approach of applying the one-dimensional wavelet transform twice at each scale [30], once horizontally and once vertically. However, we depart from the usual four-band division of the frequency plane (high-high, high-low, low-high, low-low) shown in Figure 7(a). If we followed this division, the minimizations at each node of the dependency tree (29) would depend on four variables: $ll^{(j)}_{m,n}$, $hl^{(j)}_{m,n}$, $lh^{(j)}_{m,n}$ and $hh^{(j)}_{m,n}$. Therefore, the complexity of each minimization would grow from $O(N_d^2)$ to $O(N_d^4)$, which is only feasible when few disparity values are allowed.

Instead we propose to divide the frequency plane into only three bands at each scale, as shown in Figure 7(b). The first transform is applied vertically, leading to two bands (low, high). The second transform is applied horizontally, but only onto the previous low band. This way the complexity at each node of the dependency tree remains $O(N_d^2)$, with a total complexity of $O(N_m N_n N_d^2)$ for the bottom-up pass.

F. Bitrate Optimization

The parameter b of the Laplace distribution is estimated using bracketing and a search akin to bisection [13, 28]. A large bracket is initially chosen, whose size is iteratively reduced. At the i^{th} iteration, the optimal coefficients $\{l, h\}$ are found and the actual parameter $b^{(i)}$ is estimated by minimizing the Kullback-Leibler divergence [33] between the histogram of the coefficients $\{h\}$ and the Laplace distribution (Equation 17). The current Lagrange multiplier $\lambda^{(i)}$ is obtained using the equation

$$\lambda^{(i)} = \mu^{(i)} b^{(i)} \log 2 \tag{33}$$

and the parameter μ is updated by

$$\mu^{(i+1)} = \frac{\lambda}{\lambda^{(i)}} \mu^{(i)} \tag{34}$$

where λ is the target RD slope used to encode the reference view. This update equation has the advantage of being independent of the bracket size, derivative-free, and exact when λ is a linear function of μ. The iterations end when the relative error $|\lambda - \lambda^{(i)}|/\lambda$ becomes small enough.

The final bitstream of the disparity map is generated by fixed-length coding of the low-pass coefficients in d, fixed-length coding of the sign of the high-pass coefficients and arithmetic coding [33] of their absolute values. Only the high-pass coefficients for which $\sigma(\mathsf{c})$ is one are encoded.

G. Quality Scalability

The wavelet-based encoding of the reference views allows both resolution and quality scalabilities [24]. As is, the proposed wavelet-based encoding of the disparity map only allows resolution scalability. Quality scalability is achieved by introducing quality layers [24].

The q^{th} quality layer is associated with a vector of wavelet coefficients $\mathsf{d}^{(q)}$, which is encoded using Differential Pulse Code Modulation (DPCM) between quality layers. The optimization problem then becomes

$$\min_{\{\mathsf{d}^{(q)}\}} \sum_{q=1}^{N_q} \left(\frac{1}{N} \sum_{v=0}^{N_v-1} \left\| \mathcal{M}_v^b(\mathsf{I}_v; \mathcal{T}(\mathsf{d}^{(q)})) - \hat{\mathsf{I}}_0^{(q)} \right\|_2^2 + \lambda^{(q)} R(\mathsf{d}^{(q)} - \mathsf{d}^{(q-1)} | \mathsf{c}^{(q)}, \mathsf{d}^{(q-1)}) \right) \tag{35}$$

where N_q is the number of quality layers, $\hat{\mathsf{I}}_0^{(q)}$ is the quantized reference view from the q^{th} quality layer and $\lambda^{(q)}$ is the associated Lagrange multiplier. The vector $\mathsf{d}^{(0)}$ is chosen to be the null vector. The differential vectors $\mathsf{d}^{(q)} - \mathsf{d}^{(q-1)}$ are assumed to be jointly independent and to follow a discrete and truncated Laplace distribution parameterized by $b^{(q)}$.

The optimization problem is solved sequentially for each $\mathsf{d}^{(q)}$. The minimization for the q^{th} quality layer is given by

$$\min_{\mathsf{d}^{(q)}} \frac{1}{N} \sum_{v=0}^{N_v-1} \left\| \mathcal{M}_v^b(\mathsf{I}_v; \mathcal{T}(\mathsf{d}^{(q)})) - \hat{\mathsf{I}}_0^{(q)} \right\|_2^2 + \lambda^{(q)} R(\mathsf{d}^{(q)} - \mathsf{d}^{(q-1)} | \mathsf{c}^{(q)}, \mathsf{d}^{(q-1)}) \tag{36}$$

which is similar to the minimizations described in the previous sections and can be solved in the same way.

IV. Experimental results

We present experimental results on two image sets, Tsukuba and Teddy [19], displayed in Figure 8. In both cases, the images are rectified. Note that the proposed RD optimization framework is the same whether images are rectified or not. The Tsukuba set has a fairly limited range of disparities, with only 16 disparity numbers, i.e., pixel shifts due to view changes are up to 16. On the other hand, the Teddy set has a much larger range, with 60 disparity numbers. As a consequence, the Teddy set contains much larger areas of occlusions and disocclusions.

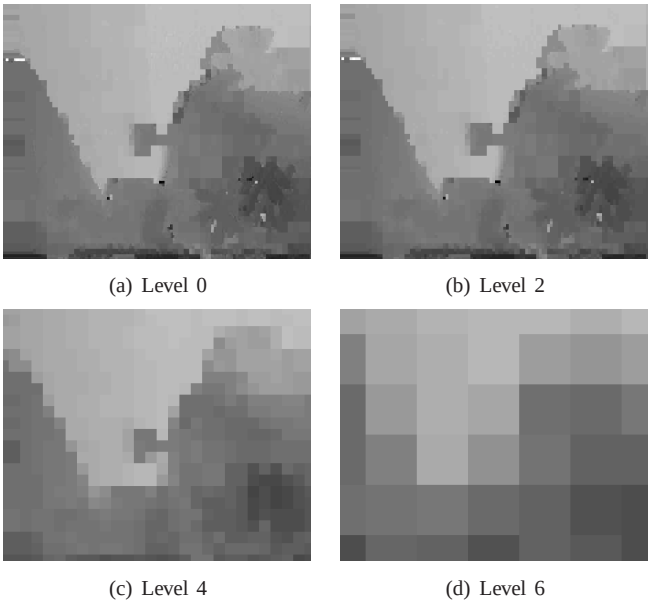
Experiments have been run in the grayscale domain with intensity values in the range $[0, 1]$. A border of two pixels has been removed around the images of Tsukuba to compensate for camera artifacts. The experiments have been conducted using nine views with the central view as reference for the Tsukuba set, and two views with the left view as reference for the Teddy set. In the following, the bitrate is defined in bits per reference-view pixels, which does not depend on the total number of views in the optimization.

In order to benchmark the performances of the proposed RD-optimized wavelet codecs, they are compared against two other classical codecs, one based on block matching [6, 10] and the other on quadtrees [23, 35]. These two codecs usually handle 2D motion vectors instead of 1D disparities. To obtain a fair comparison, they are adapted to perform one-dimensional optimizations. The encoding is performed in closed loop to obtain the least possible distortion at the decoder. The block-based encoder simply minimizes the MSE of 8×8 blocks and generates the bitstream using fixed-length codes. The quadtree-based encoder performs a full RD-optimization with variable-size blocks, as detailed in [23]. The amount of disparity regularization introduced therefore depends on the bitrate.

(a) Tsukuba image set

(b) Teddy image set

Fig. 8. The two sets of images used in the experiments (from [19]).



(a) Level 0 (b) Level 2

(c) Level 4 (d) Level 6

Fig. 9. Disparity map of the Teddy set at four resolution levels, showing the resolution scalability of the wavelet-based representation.

All codecs rely on the QccPack implementation of SPIHT [36] to encode the reference view. Therefore, the wavelets, quadtrees and blocks are all optimized using the same error tensors $\mathbf{E}$. The codecs based on quadtrees and wavelets allocate automatically the bitrate of disparity maps. Therefore, the codecs are compared at RD points with equal RD slopes, but possibly different total bitrates.

Figure 9 illustrates the resolution scalability of the proposed wavelet-based representations. Unlike quadtrees which store the disparity information only at their leaves, wavelets store this information over the entire tree, which allows partial decoding of the tree at multiple resolutions. In the experiments, the wavelet decomposition is performed completely, that is, until the low-pass band is reduced to a unique pixel. Experiments have shown that stopping the decomposition earlier, as is usually done in image coding, does not allow enough information aggregation in large textureless regions and leads to erroneous disparity estimations.

Figures 10 and 11 show the DIBR encoded at three RD slopes λ, approximately 1×10^{-2}, 2×10^{-3} and 4×10^{-4}, which correspond to reference views encoded at bitrates of 0.1bpp, 0.5bpp and 1.0bpp.

The block-based encoder appears extremely sensitive to the lack of image texture. At low bitrates, the disparity map becomes extremely noisy and is a poor estimation of the ground truth. The noise is much reduced at higher bitrates, but remains significant in some areas like the upper-right corner of Tsukuba or the roof of the house in Teddy. This seriously hinders the synthesis of novel viewpoints.

The quadtree-based encoder proves to be much more reliable. Using the RD-optimization, it is able to gracefully decrease the quality of the disparity map when the bitrate is reduced. Not only do the disparity maps become coarser, but they also tend to have less spurious noise because such a noise has a high bitrate cost.

The wavelet-based encoders, using both the S transform and the L transform, demonstrate a similar behavior. Compared to the quadtree-based encoder, they tend to generate disparity maps with less spurious noise. In quadtrees, the rate constraint favors larger blocks. However, the disparity values between blocks are independent. In wavelets, on the other hand, the rate constraint favors small wavelet coefficients, which creates dependencies between blocks and enforces an inter-block smoothness. The superiority of wavelets over quadtrees is especially noticeable in the case of larger disparity ranges, which makes them more effective at estimating and encoding the complex geometry of realistic 3D scenes.

All of these encoders have issues in areas of occlusions and disocclusions, as can be seen for instance around the chimney of the Teddy set. This creates large disparity errors which are detrimental for novel-view synthesis. This issue is confirmed by Figure 12. It shows two views synthesized from the DIBR encoded at the RD slope 2×10^{-3}, along with the differences between the synthesized and actual views. The dominant noise is due to occlusions and disocclusions. It has two sources. First, in these areas there are no correspondences between images, which leads to erroneous disparity estimations. Second, the hole-filling process is efficient when disocclusions are small but has difficulties handling large occlusions like the one on the right of the Teddy set.

We confirm this qualitative analysis by a quantitative one. Figure 13 shows the RD performances of all the codecs. The block-based codec is the least efficient. Without some kind

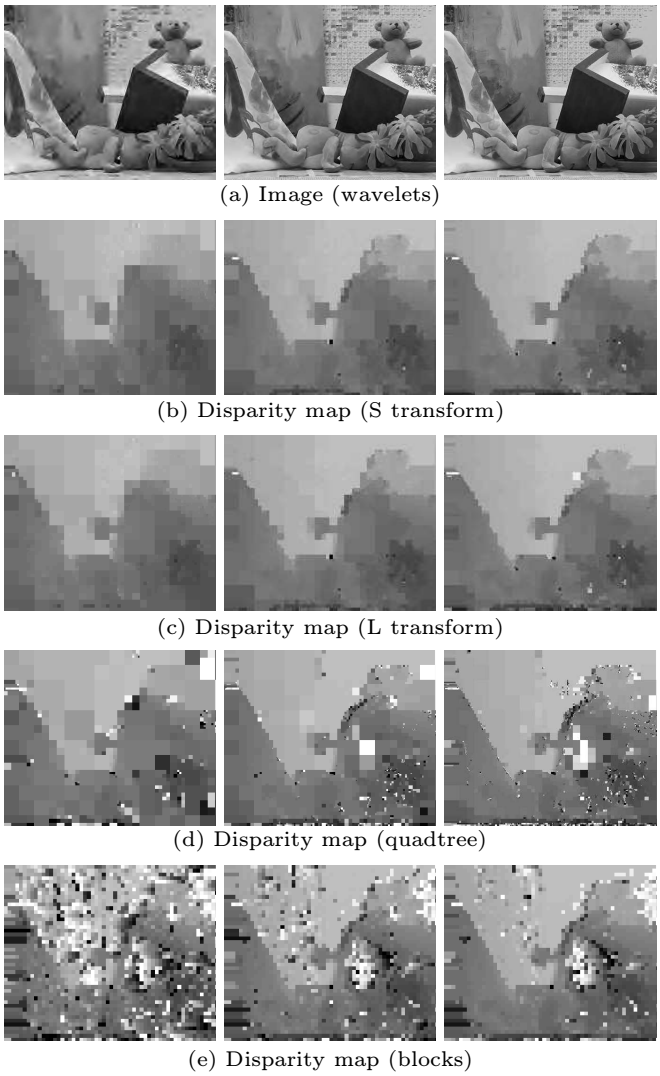


Fig. 10. The DIBR of the Teddy set at three RD slopes corresponding to reference-view bitrates of 0.1bpp, 0.5bpp and 1.0bpp (from left to right). The S and L transforms generate disparity maps which degrade gracefully with the bitrate and contain less spurious noise than quadtrees or blocks.

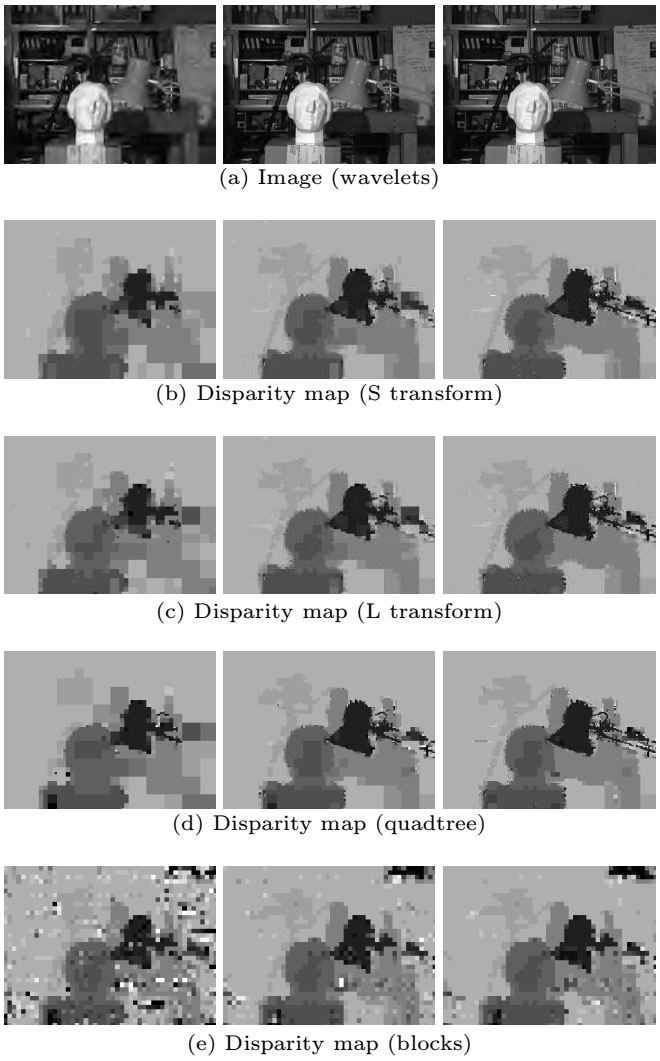


Fig. 11. The DIBR of the Tsukuba set at three RD slopes corresponding to reference-view bitrates of 0.1bpp, 0.5bpp and 1.0bpp (from left to right). The S and L transforms generate disparity maps which degrade gracefully with the bitrate and contain less spurious noise than quadtrees or blocks.

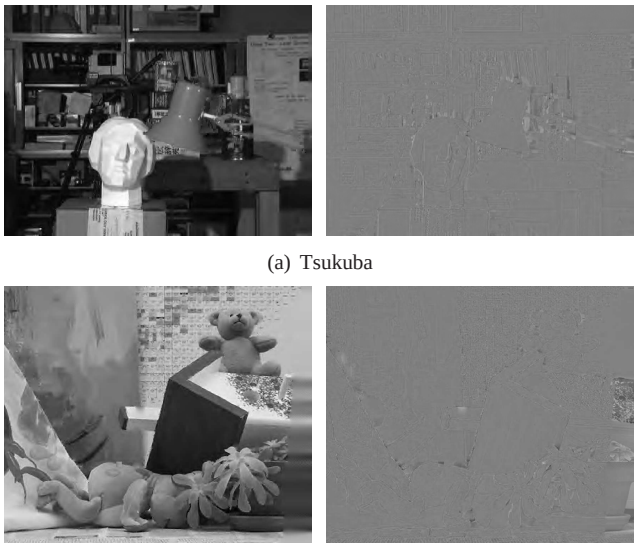
of regularization this method is not suitable for novel view synthesis, which underlines the interest of jointly estimating and encoding the disparity map. In Tsukuba, where the disparity range is small, quadtrees outperform the L transform by up to 0.09dB and the S transform by up to 0.12dB. On the other hand, in Teddy, where the disparity range is much larger, the wavelets outperform quadtrees by up to 0.84dB for the L transform and up to 0.70dB for the S transform at high bitrate. Both the L and the S transform offer similar RD performances. The advantage of the L transform is primarily the lower computational complexity of its optimization.

Figure 14 compares the quality-scalable versions of the wavelets to their non-scalable counterpart. In Tsukuba, quality scalability has a PSNR cost of at most 0.29dB at high bitrate, both for the S and L transform. In Teddy, the PSNR cost is lower for the L transform, with at most 0.34dB at high bitrate, than for the S transform, with at most 0.47dB at high bitrate.

Finally, Figure 15 reports the optimized bitrate allocation between the reference view and the disparity map. In these experiments, the allocation remains stable across most of the range of bitrates, with between 13% and 23% of the total bitrate dedicated to the disparity map. This is consistent with the heuristic ratio of 10% proposed in [9]. The allocation is similar whether the baseline is small, like in Tsukuba, or reasonably large, like in Teddy.

V. Conclusion

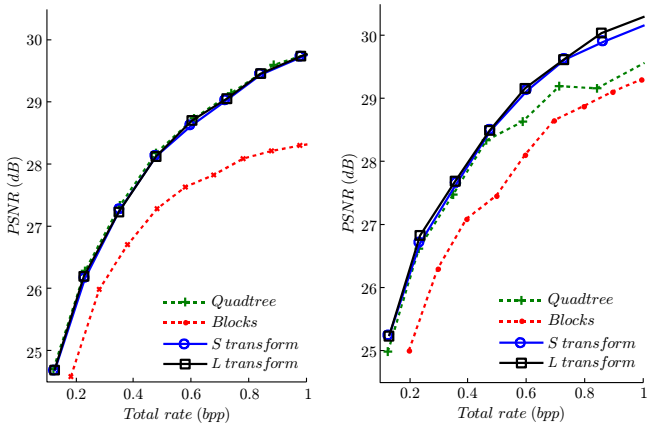
This paper has proposed a novel wavelet-domain DIBR codec able to approximate static plenoptic functions locally. The wavelet coefficients for both the images and the disparity maps have been estimated and encoded jointly to provide an optimized bitrate allocation and reduce the ambiguity of the disparity estimation. In spite of the non-linearity of the optimization problem, a globally-optimal encoding of the disparity maps has been found using dynamic programming along the tree of integer wavelet coefficients. In addition to the resolution



(a) Tsukuba

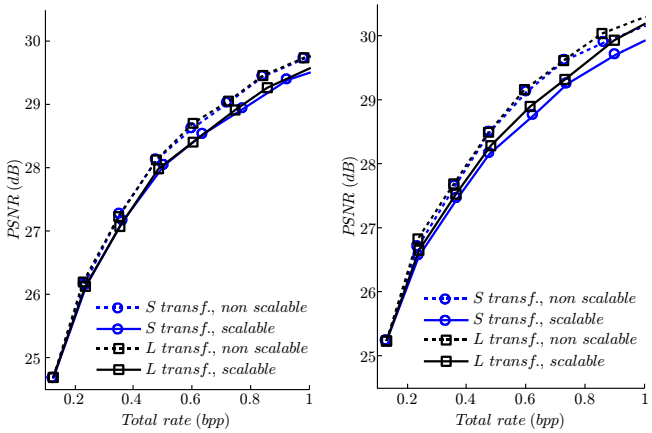
(b) Teddy

Fig. 12. Views synthesized from the DIBR with a reference view encoded at 0.5bpp (left) and differences with the original views (right). At low quantization noise, the errors are mostly due to occlusions.



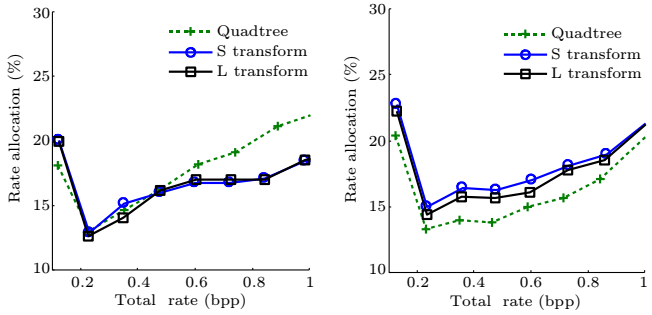
(a) Tsukuba (b) Teddy

Fig. 13. Rate-distortion performances of the encoders based on wavelets (S and L transforms), quadtrees and blocks. Wavelets are superior to quadtrees and blocks in the case of larger disparity ranges.



(a) Tsukuba (b) Teddy

Fig. 14. RD loss due to quality-scalable coding. The loss remains limited over the whole range of bitrates.



(a) Tsukuba (b) Teddy

Fig. 15. Fraction of the bitrate allocated to the disparity maps. Except at very low bitrates, the rate ratios are stable with values between 13% and 23%.

scalability intrinsic to wavelets, quality scalability has been introduced using quality layers. Finally, experimental results on real data have confirmed the performances of the proposed codec. Future work shall aim at extending the optimization of the disparity map to more general integer wavelets, at mitigating the issues due to occlusions and at compressing dynamic plenoptic functions.

REFERENCES

[1] C. Fehn, R. Barre, and R. S. Pastoor, "Interactive 3-D TV – concepts and key technologies," *Proc. of the IEEE*, vol. 94, no. 3, pp. 524–538, 2006.

[2] W. Matusik and H. Pfister, "3D-TV: A scalable system for real-time acquisition, transmission, and autostereoscopic display," in *SIGGRAPH*, 2004.

[3] E. Adelson and J. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*, M. S. Landy and J. A. Movshon, Eds. MIT Press Press, 1991, pp. 3–20.

[4] H.-Y. Shum, S. B. Kang, and S.-C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 13, pp. 1020–1037, 2003.

[5] M. Magnor, P. Ramanathan, and B. Girod, "Multi-view coding for image-based rendering using 3-D scene geometry," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 13, no. 11, pp. 1092–1106, 2003.

[6] S.-C. Chan, K.-T. Ng, Z.-F. Gan, K.-L. Chan, and H.-Y. Shum, "The plenoptic video," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 15, no. 12, pp. 1650–1659, 2005.

[7] C. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *SIGGRAPH*, 2004.

[8] L. Levkovich-Maslyuk, A. Ignatenko, A. Zhirkov, A. Konushin, I. K. Park, M. Han, and Y. Bayakovski, "Depth image-based representation and compression for static and animated 3-D objects," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 14, pp. 1032–1045, 2004.

[9] A. Smolic and P. Kauff, "Interactive 3-D video representation and coding," *Proc. of the IEEE*, vol. 93, no. 1, pp. 98–110, 2005.

[10] J. Oh, Y.-S. Choi, R.-H. Park, J. Kim, T. Kim, and H. Jung, "Trinocular stereo sequence coding based on MPEG-2," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 15, no. 3, pp. 425–429, 2005.

[11] R.-S. Wang and Y. Wang, "Multiview video sequence analysis, compression, and virtual viewpoint synthesis," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 10, pp. 397–410, 2000.

[12] R. Balter, P. Gioia, and L. Morin, "Scalable and efficient coding using 3D modeling," *IEEE Trans. on Multimedia*, vol. 8, pp. 1147–1155, 2006.

[13] A. Ortega and K. Ramchandran, "Rate distortion methods in image and video compression," *IEEE Signal Proc. Mag.*, 1998.

[14] P. Ramanathan and B. Girod, "Rate-distortion analysis for light field coding and streaming," *EURASIP Sig. Proc.: Im. Com.*, vol. 21, pp. 462–475, 2006.

[15] J. Park and H. Park, "A mesh-based disparity representation method for view interpolation and stereo image compression," *IEEE Trans. on Image Proc.*, vol. 15, pp. 1751–1762, 2006.

[16] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena scientific, 2005.

[17] D. Tzovaras and M. G. Strintzis, "Motion and disparity field estimation using rate-distortion optimization," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 8, pp. 171–180, 1998.

[18] J. Ellinas and M. Sangriotis, "Stereo video coding based on quad-tree decomposition of B-P frames by motion and disparity interpolation," *IEE Proc.-Vis. Im. Sig. Proc.*, vol. 152, no. 5, pp. 639–647, 2005.

[19] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. of Comp. Vis*, vol. 47, no. 1–3, pp. 7–42, 2002.

[20] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Proc. Mag.*, pp. 74–90, 1998.

[21] Y. Yang and S. S. Hemami, "Generalized rate-distortion optimization for motion-compensated video coders," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 10, pp. 942–955, 2000.

[22] G. M. Schuster and A. K. Katsaggelos, "An optimal quadtree-based motion estimation and motion-compensated interpolation scheme for video compression," *IEEE Trans. on Image Proc.*, vol. 7, pp. 1505–1523, 1998.

[23] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of images and video," *IEEE Trans. on Image Proc.*, vol. 3, pp. 327–331, 1994.

[24] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Springer-Verlag, 2001.

[25] M. Maitre, Y. Shinagawa, and M. N. Do, "Rate-distortion optimal depth maps in the wavelet domain," in *Proc. ICIP*, 2007.

[26] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[27] M. Woo, J. Neider, T. Davis, and Shreiner, *OpenGL programming guide*, 3rd ed. Addison Wesley, 1999.

[28] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, 1993.

[29] D. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Kluwer Academic Publishers, August 2003.

[30] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[31] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. on Com.*, vol. COM-31, no. 4, pp. 532–540, 1983.

[32] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Info. Theory*, vol. 47, pp. 498–519, 2001.

[33] T. Cover and J. Thomas, *Elements of Information Theory*. John Wiley and Sons Ltd, 1991.

[34] P. Felzenszwalb and D. Huttenlocher, "Efficient belief propagation for early vision," in *Comp. Vision and Pattern Recog.*, 2004.

[35] J. D. Oh and R.-H. Park, "Reconstruction of intermediate views from stereoscopic images using disparity vectors estimated by the geometrical constraint," *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 16, pp. 638–641, 2006.

[36] J. E. Fowler, "QccPack: an open-source software library for quantization, compression, and coding," in *Proc. of SPIE Appli. of Digital Im. Proc.*, 2000.