Inter-View Wavelet Compression of Light Fields with Disparity-Compensated Lifting

Chuo-Ling Chang, Xiaoqing Zhu, Prashant Ramanathan and Bernd Girod*

Information Systems Laboratory, Department of Electrical Engineering
Stanford University, USA

Invited Paper

ABSTRACT

We propose a novel approach that uses *disparity-compensated lifting* for wavelet compression of light fields. Disparity compensation is incorporated into the lifting structure for the transform across the views to solve the irreversibility limitation in previous wavelet coding schemes. With this approach, we obtain the benefits of wavelet coding, such as scalability in all dimensions, as well as superior compression performance. For light fields of an object, shape adaptation is adopted to improve the compression efficiency and visual quality of reconstructed images.

In this work we extend the scheme to handle light fields with arbitrary camera arrangements. A view-sequencing algorithm is developed to encode the images. Experimental results show that the proposed scheme outperforms existing light field compression techniques in terms of compression efficiency and visual quality of the reconstructed views.

Keywords: light field, multi-view image, compression, disparity compensation, lifting

1. INTRODUCTION

Image-based rendering has emerged as an important new alternative to traditional image synthesis techniques in computer graphics. With image-based rendering, scenes can be rendered by sampling previously acquired image data, instead of synthesizing from light and surface shading models and scene geometry. A *light field*[1, 2] is a data set for image-based rendering. It captures the outgoing radiance from a particular scene or object, at all points in 3-D space and in all directions. In practice, light fields are usually represented as a set of views in 2-D images, together with camera calibration information for each view.

The major objective of light field compression is to fully exploit the *intra-view* and *inter-view* coherence in the data set: intra-view refers to the relationship among pixels within the same view, and inter-view refers to the relationship between pixels in views captured from different view-points. In addition, it is desirable to have a scalable representation of the light field, which allows the system to efficiently adapt to varying resources by decompressing and rendering the light field only up to a certain resolution, quality, or bit-rate requirement.

An early light field compression algorithm employs vector quantization (VQ) to exploit the inter- and intra-view coherence.[1] The discrete wavelet transform (DWT) has also been proposed to exploit the coherence as well as to achieve scalability. Magnor et al.,[3] for instance, apply the 4-D Haar transform directly to the 4-D light field data set, followed by a 4-D extension of the *Set Partitioning in Hierarchical Trees* (SPIHT) algorithm.[4] Due to parallax, however, a point from the target scene appears at different pixel locations in different views, therefore the inter-view coherence is not fully utilized, resulting in fairly low compression efficiency.

A commonly used technique in light field compression to account for such discrepancies in view-points is referred to as *disparity compensation*, akin to motion compensation in video coding. It is used in some DPCM-like prediction-based coders,[5] which have good compression efficiency but only provide limited support of scalability. In 6, disparity compensation is incorporated into a scalable coder by applying the 4-D wavelet coder[3] to an aligned re-parametrization of the views based on an explicit geometry model. The problem with the

*{chuoling,zhuxq,pramanat,bgirod}@Stanford.EDU

scalable coder, however, is that the resampling process involved is irreversible and introduces degradation in image quality.

This mirrors recent work in 3-D subband coding of video. Many attempts have been made to incorporate motion compensation into the 3-D subband coding framework. Earlier works are somewhat unsatisfactory either for reasons similar to the resampling process in light field compression, or because the displacement vector field is severely restricted. Recently, a technique called *motion-compensated lifting*[7–9] has been proposed, which successfully incorporates unrestricted motion compensation into 3-D subband coding in a reversible fashion.

For light fields represented as a 2-D array of camera views, we have proposed a wavelet coding scheme that achieves reversibility using *disparity compensated lifting*,[10] a technique analogous to motion-compensated lifting for video. For light fields describing an object with extraneous background, shape adaptation is further proposed to improve compression efficiency and reconstruction quality.[11] In this paper, we summarize the light field compression scheme using disparity-compensated lifting and shape adaptation. Additionally, we extend the scheme to handle a more general representation of light fields, namely unstructured light fields,[12] allowing arbitrary camera positions instead of restricting the views to a 2-D grid.

The remainder of the paper is organized as follows. In Section 2, we discuss prior work using disparity compensation for light field compression and their limitations. In Section 3, we present the proposed algorithm of sequencing the unstructured camera views of general light fields, as preparation for the inter-view wavelet transform. We explain the key idea of disparity-compensated lifting for carrying out the transform in Section 4 and describe coding of the subsequent subband images in Section 5, as two major stages of the system. Extensions for shape adaptation are discussed in Section 6. Experimental results are given in Section 7.

2. PRIOR WORK WITH DISPARITY COMPENSATION

Disparity compensation is originally proposed for stereo and multi-view image compression,[13,14] and is also extensively used in compression of concentric mosaics, a 3-D data set for image-based rendering.[15,16] Most light field compression approaches incorporate some form of disparity compensation for compression efficiency.

For light fields, a geometry proxy can be used to facilitate disparity compensation.[5] A geometry model of the scene is first estimated from the acquired camera views using, for instance, computer vision techniques.[17] With the geometry model, a point in one view can be associated to its corresponding points, i.e., points referring to the same 3-D position of the scene, in other views. Therefore, the geometry model directly provides a dense disparity map between any pair of the views in both directions. Furthermore, the geometry model can also be used to improve the rendering quality.[2,12]

To encode the views, Magnor et al. describe a geometry-based disparity-compensated predictive method with block-wise discrete cosine transform (DCT) coding of the residual.[5] Multiple reference views are used to predict each view, allowing multiple coding modes for each block in the view. The scheme is later extended to incorporate multiple hypothesis in the selection of reference views.[18] These prediction-based coders achieve good compression performance; however, the support for scalability is limited.

Another way to incorporate the geometry for disparity compensation is to re-parameterize all the views to a common reference frame using the geometry model. In 6, a texture-map based approach is proposed. A geometry model is first estimated from the light field.[17] The views in the data set are warped onto the geometry, generating a set of aligned view-dependent texture-maps. These texture-maps are then coded by the 4-D Haar transform and the 4-D SPIHT algorithm. The 4-D transform effectively exploits the coherence along all dimensions, meanwhile a scalable representation is naturally provided by the Haar transform and SPIHT coding.

To reconstruct the views, however, the reconstructed texture-maps need to be projected back to their original view-points. If portions of the image were contracted during the warping process, there is a permanent loss in resolution. In addition, the interpolation involved in the procedure is usually not reversible. As a result, the reconstruction of the views not only exhibits the quantization noise in the wavelet coefficients, but also inevitably inherits the distortion caused by the warping process. Moreover, the approach can only encode the portions of the view covered by the geometry model because of the underlying parametrization. The quality degradation due to resampling (warping) can affect the quality of the rendered view.[12] Accordingly, for light field compression,

our objective is to minimize the distortion in the reconstruction of the acquired camera views for a given bit-rate constraint.

In the following sections, we describe a novel approach to solve the problems caused by the resampling process. Disparity compensation is effectively incorporated into the inter-view wavelet transform so that reconstruction quality of the views is only affected by coefficient quantization. Additionally, in contrast to prediction-based coders, scalability is naturally supported by the DWT in the proposed scheme.

3. VIEW SEQUENCING

In order to apply the inter-view wavelet transform, the images in the data set need to be organized as a sequence of views, with a specific scanning order for compression. This view sequence should have the property that neighboring views in the sequence exhibit higher coherence than views that are further apart, so that the wavelet transform can decorrelate the signals more effectively.

Previously, we have considered compression of light fields acquired by cameras positioned as a 2-D array (on a hemispherical surface, in polar coordinates) encompassing the scene.[10] The data sets can therefore be easily represented as a 2-D array of camera views. In such a case, the columns and the rows in the array structure naturally form the view sequence required for the wavelet transform. The inter-view transform is carried out by applying 1-D transforms horizontally and vertically across the 2-D array, resulting in a 2-D inter-view transform.

Not all light field data sets, however, bear such a simple structure. Some light fields are captured by hand-held cameras moving around the scene.[12] Many others have denser samples of views for a particular part of the scene in order to capture more details for the part of interest. In these so-called unstructured light fields[12] where the cameras are not positioned on a regular grid, we wish to retain the property that neighboring views in the sequence exhibit higher coherence.

We propose to formulate the view sequencing problem of unstructured light fields as the *Travelling Salesman Problem (TSP)*,[19] i.e., finding the cheapest closed tour visiting a set of nodes, starting from a node, visiting every node exactly once and returning to the initial node. Based on the assumption that the views taken from nearby cameras have higher coherence, we calculate the cost between two views as the Euclidian distance between their corresponding cameras. As a result, the goal of view sequencing becomes finding the shortest path connecting all camera positions and returning to the initial position. The corresponding camera views in this path constitute the desired view sequence. Note that returning to the initial position is not required for the view sequencing problem. However, this enables us to adopt existing algorithms developed for solving TSP.

In addition to the camera distance, other metrics such as the similarity of viewing directions and image resolution also account for the coherence between two views.[12] In this work, we assume that all cameras are approximately looking at the center of the scene and the image resolution in all views are identical. Hence only the camera distances are of concern. In general, other metrics can be incorporated into the cost calculation.

Although the TSP is NP-complete, we can effectively find an approximate solution. Note that the optimal solution of TSP does not necessarily guarantee the best compression performance. Therefore, a sub-optimal solution may suffice to serve the purpose of systematically arranging the data sets into a view sequence so that the inter-view wavelet transform can be carried out efficiently.

Since the Euclidian distance is a symmetric metric, we can adopt the algorithms proposed for symmetric TSP. Specifically, we use a strategy based on Lagrangian relaxation.[19,20] As shown in Fig. 3, results of the algorithm tend to link close-by views as neighbors.

4. INTER-VIEW TRANSFORM USING DISPARITY-COMPENSATED LIFTING

Given the sequence of views, the proposed wavelet compression scheme for light fields consists of two main stages. The first stage is the inter-view transform, i.e., the wavelet transform that exploits the coherence between different views in the data set. After this, the resulting subband images still exhibit coherence among neighboring pixels. The second stage, namely coding of the subband images, is then responsible for exploiting the remaining coherence and generating the final scalable bit-stream. In this section, we introduce the inter-view transform using disparity-compensated lifting. Coding of the subband images are discussed in the next section.

4.1. Disparity Compensated Lifting

Lifting is a procedure that can be used to implement discrete wavelet transforms.[21] Suppose that, in the context of light field compression, we have a sequence of N views, $x[n]$, $n = 0, \cdots, N-1$. Assuming N is even for simplicity, we split up this set into two sets of $\frac{N}{2}$ views: an even set $x_0[k]$, $k = 0, \cdots, \frac{N}{2} - 1$, and an odd set $x_1[k]$, $k = 0, \cdots, \frac{N}{2} - 1$. Wavelet analysis can be factorized into one or more lifting steps, each consisting of a prediction and an update filter. The lifting structure transforms $x_0[k]$ and $x_1[k]$ into $y_0[k]$ and $y_1[k]$, the low-pass and the high-pass subbands resulting from the DWT of $x[n]$ respectively.

For reconstruction, as long as the filters used in wavelet synthesis are identical to those in wavelet analysis, the reversibility of the transform is ensured. We can use any kind of filters in lifting, including non-linear or data-adaptive filters, while still preserving the reversibility. For light field compression, we incorporate the geometry-based disparity compensation as discussed in Section 2 into the prediction and update filters.

Let $v_0[k]$ and $v_1[k]$ denote the view-point, i.e. the viewing position and direction, of $x_0[k]$ and $x_1[k]$ respectively. Let $w_{01}^{(k)}$ be the function that warps its input, either an even-view $x_0[k]$ or a low-pass subband image $y_0[k]$, from view-point $v_0[k]$ to $v_1[k]$ using the disparity information. Similarly, $w_{10}^{(k)}$ warps its input, either an odd-view $x_1[k]$ or a high-pass subband image $y_1[k]$, from view-point $v_1[k]$ to $v_0[k]$. As an example, $w_{01}^{(k)}(x_0[k])$ denotes the warped view (with view-point $v_1[k]$), derived from the given view $x_0[k]$ (with view-point $v_0[k]$).

To calculate one particular pixel value at location p_1 on $w_{01}^{(k)}(x_0[k])$, p_1 is first back-projected to 3-D space, from $v_1[k]$, to find the corresponding point on the geometry surface. This 3-D point is then projected to location p_0 on the image plane at $v_0[k]$. The pixel value at p_0 is then extracted from $x_0[k]$ using bilinear interpolation and assigned to p_1 on $w_{01}^{(k)}(x_0[k])$.

The disparity-compensated lifting approach uses the warping functions, $w_{01}^{(k)}$ and $w_{10}^{(k)}$, as the first stage of the prediction and update filters, respectively. For the Haar wavelet, disparity-compensated lifting can be described by the following equations:

$$y_1[k] = x_1[k] - w_{01}^{(k)}(x_0[k]) \tag{1a}$$

$$y_0[k] = x_0[k] + \tfrac{1}{2} w_{10}^{(k)}(y_1[k]) = (x_0[k] - \tfrac{1}{2} w_{10}^{(k)}(w_{01}^{(k)}(x_0[k]))) + \tfrac{1}{2} w_{10}^{(k)}(x_1[k]) \tag{1b}$$

$$\hat{x_0}[k] = y_0[k] - \tfrac{1}{2} w_{10}^{(k)}(y_1[k]) = x_0[k] \tag{1c}$$

$$\hat{x_1}[k] = y_1[k] + w_{01}^{(k)}(x_0[k]) = x_1[k] \tag{1d}$$

Note that $y_1[k]$ needs to be computed prior to $y_0[k]$ in the lifting structure. We first generate a warped view $w_{01}^{(k)}(x_0[k])$ from $x_0[k]$ to predict $x_1[k]$. The resulting disparity-compensated prediction residual, $y_1[k]$, corresponds to the high-pass subband of the Haar wavelet. This high-pass subband is then warped and added to $x_0[k]$ in order to generate $y_0[k]$, the low-pass subband, which is approximately the disparity-compensated average of $x_0[k]$ and $x_1[k]$.

For this Haar wavelet example, only one lifting step is needed as shown in Fig.1. The prediction filter, p, and the update filter, u, consist of disparity compensation followed by a scaling of -1 and $\frac{1}{2}$, respectively. The additional scaling factors G_0 and G_1 needed to normalize the transform (Fig.1) are omitted in (1).

Note that unlike the texture-map approach which needs an explicit geometry model,[6] the lifting structure can use other methods to provide the disparity information, such as block-matching. Disparity-compensated lifting effectively incorporates disparity compensation into the DWT while maintaining the reversibility of the transform. In addition, the lifting structure also allows fully in-place calculation of the wavelet transform.[21] A memory-efficient implementation using a pipeline structure has also been proposed which is especially suitable for interactive rendering applications.[16, 22]

4.2. Wavelet Kernels

Various wavelet kernels can be implemented using lifting. In this work, the Haar wavelet and the biorthogonal Cohen-Daubechies-Feauveau 5/3 wavelet[23] are adopted because of their simplicity and effectiveness. Typically,

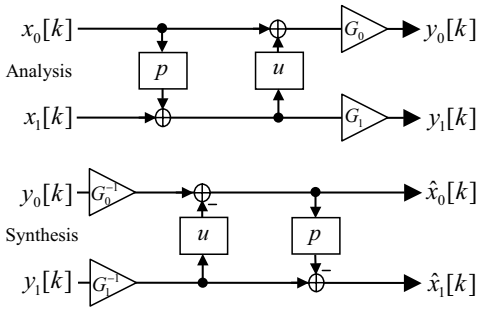

Figure 1. Lifting: wavelet analysis and synthesis using one lifting step: p is the prediction filter and u is the update filter. G_0 and G_1 are the scaling factors to normalize the transform.

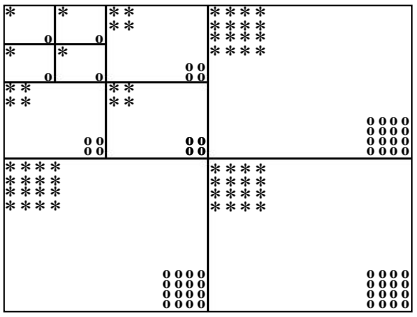

Figure 2. Block-wise SPIHT: The transform coefficients are divided into blocks. For example, the coefficients at the locations labelled by $*$ are grouped into one block, those labelled by o are grouped into another block

the 5/3 wavelet, due to its bidirectional support which enables bidirectional prediction and update, gives better performance than the Haar wavelet, at the cost of more computation. To increase coding speed, a truncated version of the wavelet kernels can also be used, in which case the low-pass subband images are replaced directly by the even views.

4.3. Multi-Level Transform

The low-pass subband image sequence $y_0[k]$ is essentially the down-sampled version of the original sequence $x[n]$, viewed at the even view-points $v_0[k]$. If the number of view-points is sufficiently large, a multi-level transform can be performed.

Instead of applying the next level of inter-view wavelet transform directly on $y_0[k]$, as in the case for wavelet coding of image and video, we propose to reorder the sequence $y_0[k]$. View sequencing (Section 3) is applied again to the views in $y_0[k]$ to generate a reordered sequence of the low-pass subband images. The inter-view transform is then applied on the reordered $y_0[k]$. This procedure can be repeated several times depending on the density and total number of the view-points.

By sequencing the views separately for each level of the inter-view transform, exploitation of the inter-view coherence is not limited in the direction determined by the view sequencing in the first level. Effectively, the direction of the wavelet transform is adaptive to the remaining coherence among views in the particular level. Although the results of view sequencing in each level need to be additionally signaled, the overhead is negligible compared to the overall size of a typical light field data set.

5. CODING OF SUBBAND IMAGES

To further compress the subband images resulting from the inter-view transform, the intra-view transform is applied followed by SPIHT coding of the wavelet coefficients and rate-distortion optimized bit-stream truncation, in order to generate the final scalable bit-stream.

5.1. Intra-View Transform

After the inter-view transform, there is remaining coherence among neighboring pixels within each subband image, especially for the low-pass subbands. To further exploit this, the intra-view transform is applied to each subband image using a multi-level 2-D DWT. The biorthogonal Cohen-Daubechies-Feauveau 9/7 wavelet,[23] popular for image compression, is chosen for the intra-view transform.

5.2. Coefficient Coding

To encode the DWT coefficients, the SPIHT algorithm is chosen for its computational simplicity and high compression efficiency.[4] It is applied to each subband image separately. For better exploitation of local statistics as well as memory efficiency, we further modify the SPIHT coder to re-group the DWT coefficients in each subband image into individual blocks and encode them separately as illustrated in Fig. 2. Similar ideas have been proposed for image compression[24] and video residual image coding.[25]

In this way, each block can be encoded starting from its own highest bit-plane and truncated at an appropriate point, as opposed to the conventional case, where the starting bit-plane and the truncation point are determined globally for the entire image. The block-wise SPIHT coder also lowers the memory requirement,[24] and allows greater freedom in light field transmission and rendering. Note that although the coefficients are coded together within each block, the intra-view wavelet transform is performed on the entire image.

Block-wise coding, however, has the overhead of signaling the index of the starting bit-plane and the truncation point for each block. It is proposed by Lin and Gray to constrain the truncation points to the end of a bit-plane so that only the index of the ending bit-plane, instead of the bitstream length, need to be coded.[25] We further propose that the end of a coding pass, i.e., the significance or refinement pass defined in the SPIHT algorithm,[4] can be included as candidate truncation points, for finer granularity at the cost of only one more signaling bit.

5.3. Rate-Distortion Optimized Bit-stream Assembly

The task of bitstream assembly is to choose the optimal truncation point for each coefficient block so as to maximize the overall reconstruction quality subject to the total bit-rate constraint, or to minimize the total bit-rate in achieving certain reconstruction quality. Using the Lagrangian multiplier technique, the constrained problem is converted to the unconstrained minimization of the Lagrangian cost function

$$J_{i,b} = D_{i,b} + \lambda R_{i,b} \tag{2}$$

where $J_{i,b}$ is the cost function in the bth block of the ith subband image, $R_{i,b}$ is the bit-rate for the block, $D_{i,b}$ is the overall reconstruction distortion introduced by the coefficient quantization in the block, and λ is the Lagrangian multiplier which corresponds to the desired tradeoff between bit-rate and reconstruction distortion.

To compute the cost function at every candidate truncation point for a coefficient block (Section 5.2), we need to obtain $R_{i,b}$ and $D_{i,b}$ at these points. To achieve this, each block is initially encoded to a pre-determined sufficiently high bit-rate. During the coding process, the bit-rate, $R_{i,b}$, at each candidate point is recorded, together with the distortion of the transform coefficients.

To compute $D_{i,b}$ efficiently, we relate the distortion of the transform coefficients, denoted as $D^t_{i,b}$, to the distortion in the pixel domain, $D_{i,b}$, based on several approximations. First, in the decoding process, the inverse intra-view transform converts the transform coefficients with distortion $D^t_{i,b}$ back to the subband images with distortion $D^s_{i,b}$. Neglecting the contributions from neighboring blocks and approximating the intra-view transform as orthogonal, we can treat $D^s_{i,b}$ and $D^t_{i,b}$ as equal. Then, ignoring the effects of disparity compensation and assuming uncorrelated distortion from different inter-view subbands as in 16, $D_{i,b}$ is approximated as a scaled version of $D^s_{i,b}$. The scaling factor is determined by the wavelet kernel, and the type (low-pass or high-pass) and level of the ith subband. It can be calculated from the wavelet synthesis filter coefficients.

For a given λ, a search is performed over all candidate truncation points for a block to find the one with the minimum Lagrangian cost $J^\lambda_{i,b}$. The bitstream is then truncated at the optimal bit-rate, $R^\lambda_{i,b}$, and the corresponding truncation point is signaled to the decoder. Note that the optimality here is subject to the constraints in truncation point selection and the approximations in distortion calculation. The same process can be repeated for different values of λ. The corresponding truncation points for the coefficient blocks at each λ are stored in a look-up-table, along with the total bit-rate and overall reconstruction distortion obtained by actually decoding the bit-stream.

For a request of transmission with a certain bit-rate or distortion requirement, the bitstream assembler simply looks for the closest operating point, finds the corresponding truncation points for each block, truncates

the bitstreams, adds auxiliary bits to indicate the truncation points, and concatenates all truncated bitstreams to form the final bitstream. If a finer granularity is required, a simple interpolation scheme can be used to obtain a closer operating point. Note that the extra burden of calculations and look-up-table storage are only at the encoder.

5.4. Scalability

In the proposed system, different reconstruction qualities can be obtained from a single encoding process by assembling the bitstreams using different truncation points. This provides reconstruction-quality scalability.

View-point scalability is supported by the inter-view wavelet transform. Specifically, the low-pass subband images are essentially the down-sampled version of the light field views, requiring a fraction of the total bit-rate. If necessary, one can decode only the low-pass subband images for rendering, as opposed to the full reconstruction that need both the low-pass and high-pass subband images to be decoded.

Moreover, the intra-view wavelet transform provides image-resolution scalability. Depending on the applications, the views in the light field can be decompressed up to the full resolution, or only a fraction of it, from a single compressed bitstream. However, for the SPIHT algorithm to achieve image-resolution scalability, i.e., to gather the bits regarding to the low-pass intra-view subbands at the beginning of the bitstream, the output order would have to be re-designed.[26]

6. SHAPE ADAPTATION

When the light field of interest represents the exterior views of a 3-D object, the constituent images contain extraneous background pixels and discontinuities at the object boundaries. Hence, we might encode unnecessary pixels, and there is increased energy in the high frequency components. Both effects are the cause of compression inefficiencies. We therefore propose to incorporate shape adaptation using the 2-D shape of the object in each view. If the projection of the geometry model used for disparity compensation is consistent with the 2-D object shapes, the geometry itself can account for the shape information. No extra shape coding is needed. Conversely, if the geometry model only provides approximate shape information, we can code the exact shape using the available approximation.[11] Note that by setting the object shape to the entire image, the shape adaptation technique can be reduced to conventional coding without loss of generality.

When shape information is available, better disparity compensation at the object boundaries can be performed. In particular, with an inaccurate geometry model, an object pixel in one view may be disparity-compensated to the background in another view. With knowledge of exact object boundaries, on the other hand, the prediction can be obtained from the nearest object pixel instead of the background.

Without shape adaptation, a significant portion of the bitstream is spent on encoding the background. With shape information, however, the Shape-Adaptive DWT (SA-DWT)[27] can be used. For each view the transform is performed on the entire image, generating as many wavelet coefficients as object pixels. In addition, since the shape-adaptive scheme avoids performing the transform across object boundaries, extraneous high frequency components are avoided, contributing to improved coding efficiency and enhanced reconstruction quality.

The SPIHT algorithm is also modified to disregard zero-tree subtrees that contain only background pixels. Specifically, whenever the children of a node contains only background pixels, the tree is terminated because there is no further information.[28] Note that, conventionally, bitstreams from SPIHT coding are further compressed by a context-based adaptive arithmetic coder, whereas with shape adaptation, there will likely be a much smaller performance gain from appending the arithmetic coder.[28] Therefore, the need of arithmetic coding is eliminated, and coding complexity can be reduced without much sacrifice in compression efficiency.

7. EXPERIMENTAL RESULTS

Experimental results are shown for two light field data sets, *Buddha* and *Bust*. Examples of the data sets are shown in Fig. 8 and Fig.9. *Buddha* is a computer synthesized data set with 280 views, each with a resolution of 512×512, together with known geometry model and camera parameters. *Bust* consists of 338 views of a real-world object; each view has a resolution of 384×768. The geometry model and camera parameters are estimated

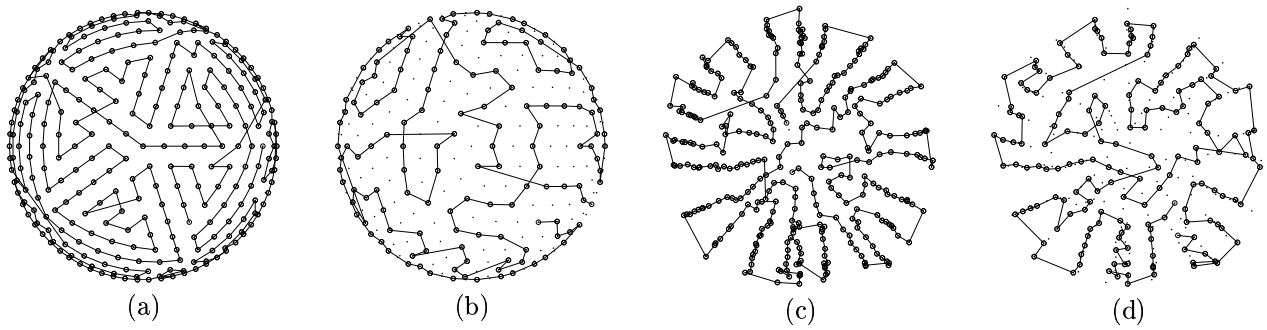

Figure 3. View sequencing for the inter-view transform: In both data sets, the cameras are approximately distributed on the hemisphere surrounding the object. The dots denote the camera positions viewing from the top of the hemisphere. The lines denote the view sequence for different levels of the inter-view transform. (a)*Buddha* 1-level (b)*Buddha* 2-level (c)*Bust* 1-level (d)*Bust* 2-level

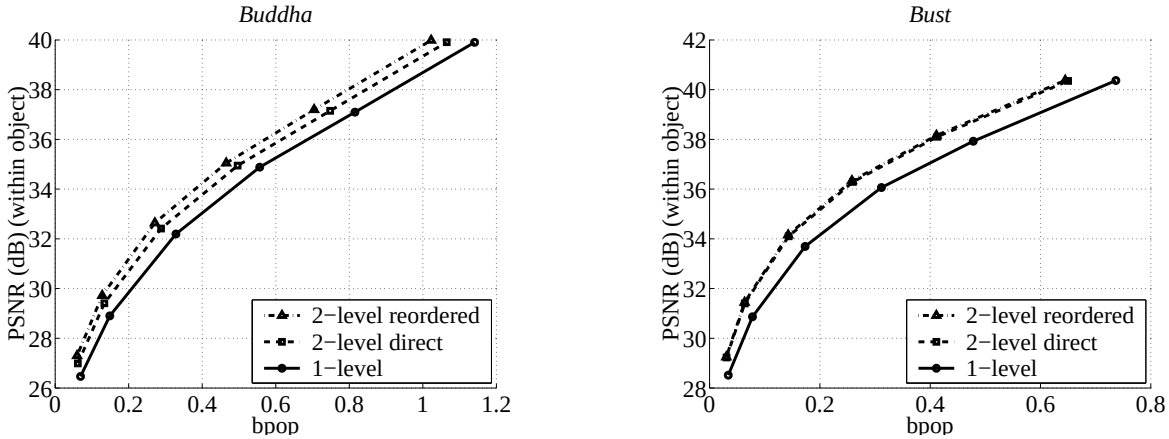

Figure 4. Rate-PSNR curves for 1- and 2-level of view-sequencing

from the acquired camera views using methods described in 29. In both data sets, the camera positions are approximately distributed on a hemisphere surrounding the object. The camera positions and the results of view sequencing, obtained from the method described in Section 3, are shown in Fig.3. In the following experiments, only the luminance component is encoded.

The geometry model and camera parameters are, in general, used for rendering as well. Therefore, the bit-rate for encoding such information is not included in the following results. For shape adaptation, the projection of the geometry model is directly used as the 2-D shape of the object. No extra coding is needed. With shape adaptation, only the object pixels are encoded. Hence, the conventional *bit-per-pixel* (bpp) measurement of bit-rate is modified to *bit-per-object-pixel* (bpop), defined as the length of the final bitstream divided by the number of object pixels in the data set. Similarly, the *Peak-Signal-to-Noise-Ratio* (PSNR) measurement for the reconstruction quality is modified to be computed by averaging over only the object pixels in all of the views. The compression performance is shown using the rate-PSNR curves, which express the relation between the bit-rate (bpop) and reconstruction quality (PSNR).

Throughout the experiments, the 5-level intra-view transform with the 9/7 wavelet is chosen as it gives the best empirical performance.

7.1. View Sequencing

The compression performance after sequencing the views for 1-level and 2-level inter-view decomposition, according to the proposed algorithm in Section 3, is presented in Fig. 4.

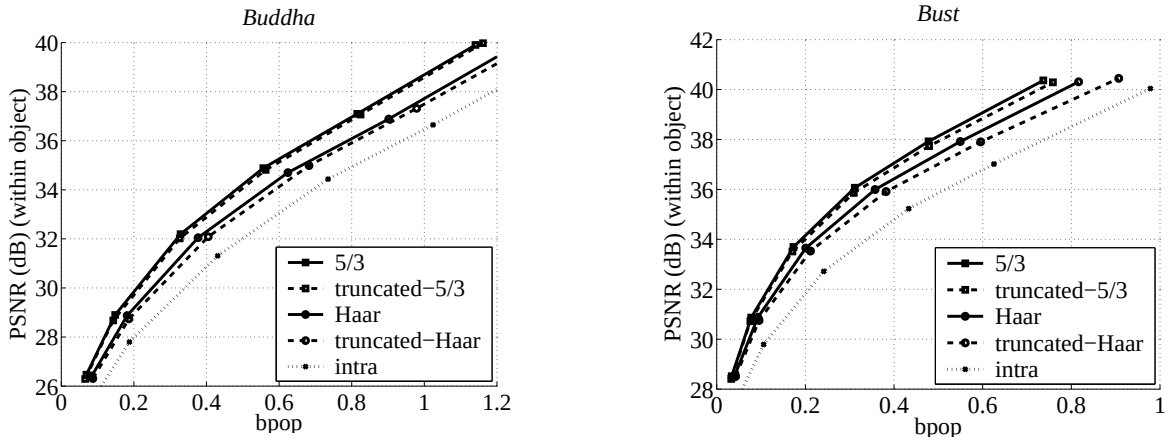

Figure 5. Rate-PSNR curves for different inter-view wavelet kernels using the 1-level inter-view transform

There are two options for sequencing the 2-level inter-view decomposition. We can simply use the even views from the 1-level view sequence in the same order, denoted as "2-level direct" in the figure. Or we can potentially improve this ordering by re-optimizing the order of the even views, denoted as "2-level re-ordered". Fig. 4 shows small improvement in the rate-PSNR performance for the "re-ordered" versus the "direct" method for the *Buddha* data set, and negligible gains for *Bust*.

7.2. Inter-View Wavelet Kernels

The compression performance of four different inter-view wavelet kernels, Haar, 5/3, truncated-Haar, and truncated-5/3, are compared together with the intra-coding scheme with no inter-view transform. The 1-level inter-view transform is used. The rate-PSNR curves are shown in Fig. 5.

At the same bit-rate, the 5/3 wavelet performs about 0.8-1.0 dB better than the Haar wavelet. The truncated kernels perform slightly worse than their non-truncated counterparts. Compared to the intra-coding scheme, the inter-view transform with the 5/3 wavelet provides a 2.0-2.2 dB gain in terms of PSNR at the same bit-rate, or equivalently a bit-rate reduction of 30%-40% for the same reconstruction quality.

7.3. Multiple Inter-View Transform Levels

In these experiments, the 5/3 wavelet is used. The results for the 1-level, 2-level, and 3-level inter-view transform are shown in Fig. 6, along with the intra-coding scheme.

For the two data sets, experiments show that there is about a 1 dB gain by applying the 2-level transform over the 1-level transform. However, the performance degrades when using the 3-level transform. This may be due to the fact that two neighboring views in the 3-level transform are too far away to allow an efficient decomposition. For data sets with denser view-points, the gain by using a multi-level inter-view transform is expected to be larger.

7.4. Comparison with Existing Techniques

We compare the proposed coder with the shape-adaptive DCT (SA-DCT) coder,[11] as an example of the state-of-the-art. A comparison with the texture-map coder[6] for other two data sets, *Garfield* and *Penguin*, can also be found in 10. The rate-PSNR curves are shown in Fig. 7. Examples of the the reconstructed views using the two coders at similar bit-rates are shown in Fig. 8 and Fig. 9.

For the proposed coder, we use 2-level of inter-view transform with the 5/3 wavelet for *Buddha* and 3-level for *Bust*, as they give the best performance from previous experiments.

The SA-DCT coder, described in greater details in 11, employs a hierarchical bi-directional predictive structure to encode all the views. Note that the compression efficiency of the SA-DCT coder largely relies on the

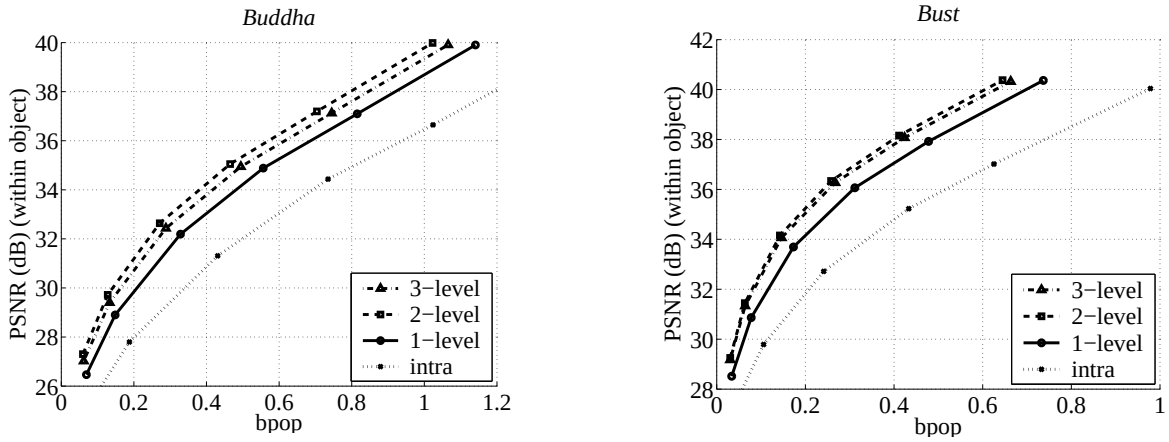

Figure 6. Rate-PSNR curves for different inter-view transform levels using the 5/3 wavelet

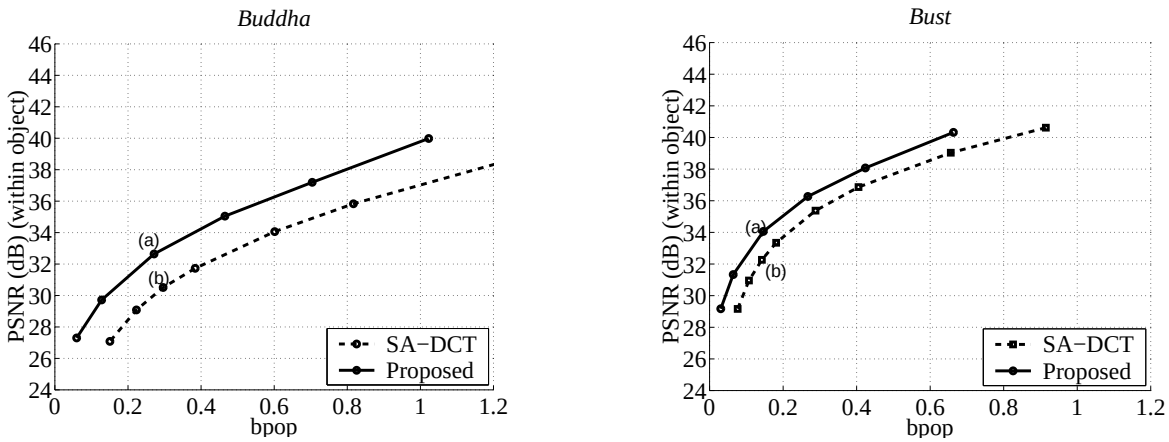

Figure 7. Comparison with the existing techniques.

prediction structure involved, and therefore can cover a wide range. To maintain the same random access capabilities for both coders, the levels of prediction are kept the same as the levels of inter-view wavelet transform in the proposed coder for both data sets. More specifically, the 3-level prediction corresponds to 1/8 of the views being intra-coded, while the 2-level prediction yields a ratio of 1/4 for intra-coded views.

From the rate-PSNR curves in Fig. 7, we can see that the proposed scheme exhibits superior compression efficiency over the SA-DCT coder. For *Buddha*, the proposed scheme outperforms the SA-DCT coder by a gain of 2 dB in terms of object PSNR, or equivalently a reduction of 30% in bit-rate. In the case of *Bust*, the gain in object PSNR for the proposed coder is 1-2 dB, or equivalently the bit-rate reduction is up to 20%. Note that these results are consistent with our previous observations.[10]

The proposed coder further achieves better visual quality in its reconstructions. As shown in Fig. 8(a) and Fig. 9(a), there are no blocking artifacts in the reconstructed views, which are observable in the magnified images of the corresponding SA-DCT coder results. The proposed coder additionally provides scalability for image resolution and reconstruction quality, unattainable by the SA-DCT coder.

8. CONCLUSIONS

We propose a novel approach for light field compression that uses disparity-compensated lifting for inter-view wavelet coding. The lifting structure effectively integrates disparity compensation into wavelet coding while preserving reversibility. Reconstruction of the acquired views is free of the distortion caused by the irreversible

Figure 8. Luminance component of *Buddha*: The reconstructed view from different coders, corresponding to the labelled points on the rate-PSNR curves in Fig. 7(a), are shown on the top row, with the white box labelling the area magnified on the bottom row. (a) proposed at 0.271 bpop (b) SA-DCT at 0.296 bpop

resampling process. The proposed scheme also supports scalability in image resolution, view-point and reconstruction quality.

The scheme is further extended to accommodate unstructured light fields, by formulating the view sequencing problem as the Travelling Salesman Problem and giving an approximate solution using existing techniques. For light fields of an object with extraneous background, shape adaptation techniques are applied to improve coding efficiency as well as visual quality of reconstruction.

Compared with the SA-DCT coder, the proposed coder exhibits superior compression efficiency, improves the support of scalability, as well as achieving better visual quality of the reconstructed views.

ACKNOWLEDGMENTS

This work was supported, in part, by a gift from Intel Corporation and, in part, by Grant No. ECS-0225315 of the National Science Foundation. The data sets used for the work, *Buddha* and *Bust*, are courtesy of Intel Research.

REFERENCES

1. M. Levoy and P. Hanrahan, "Light field rendering," in *Computer Graphics (Proceedings SIGGRAPH 96)*, pp. 31–42, August 1996.
2. S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Computer Graphics (Proceedings SIGGRAPH 96)*, pp. 43–54, August 1996.
3. M. Magnor, A. Endmann, and B. Girod, "Progressive compression and rendering of light fields," in *Proceedings Vision, Modelling and Visualization 2000*, pp. 199–203, November 2000.
4. A. Said and W. A. Pearlman, "A new fast and efficient image codec based on Set Partitioning in Hierarchical Trees," *IEEE Transactions on Circuits and Systems for Video Technology* **6**, pp. 243–250, June 1996.
5. M. Magnor, P. Eisert, and B. Girod, "Model-aided coding of multi-viewpoint image data," in *Proceedings of the IEEE International Conference on Image Processing ICIP-2000*, **2**, pp. 919–922, (Vancouver, Canada), September 2000.
6. M. Magnor and B. Girod, "Model-based coding of multi-viewpoint imagery," in *Proceedings SPIE Visual Communications and Image Processing VCIP-2000*, **1**, pp. 14–22, June 2000.

Figure 9. Luminance component of *Bust*: The reconstructed view from different coders, corresponding to the labelled points on the rate-PSNR curves in Fig. 7(b), are shown on the top row, with the white box labelling the area magnified on the bottom row. (a) proposed at 0.146 bpop (b) SA-DCT at 0.142 bpop

7. A. Secker and D. Taubman, "Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting," in *Proceedings of the IEEE International Conference on Image Processing ICIP-2001*, **2**, pp. 1029–1032, (Thessaloniki, Greece), October 2001.
8. B. P.-P. and V. Bottreau, "Three dimensional lifting schemes for motion compensated video compression," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP-2001*, **3**, pp. 1793–1796, (Salt Lake City, UT, USA), May 2001.
9. L. Luo, J. Li, S. Li, *et al.*, "Motion compensated lifting wavelet and its application in video coding," in *Proceedings of the IEEE International Conference on Multimedia and Expo 2001*, pp. 481–484, (Tokyo, Japan), August 2001.
10. B. Girod, C.-L. Chang, P. Ramanathan, and X. Zhu, "Light field compression using disparity-compensated lifting," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP-2003*, (Hong Kong, China), April 2003.
11. C.-L. Chang, X. Zhu, P. Ramanathan, and B. Girod, "Shape adaptation for light field compression," in *(submitted) IEEE Int. Conf. on Image Processing (ICIP-2003), Barcelona, Spain*, 2003.
12. C. Buehler, M. Bosse, L. McMmillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Computer Graphics (Proceedings SIGGRAPH 01)*, pp. 425–432, August 2001.
13. M. Lukacs, "Predictive coding of multi-viewpoint image sets," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP-1986*, pp. 521–524, (Tokyo, Japan), March 1986.
14. H. Aydinoğlu and M. H. H. III, "Compression of multi-view images," in *Proceedings of the IEEE International Conference on Image Processing ICIP-1994*, **2**, pp. 385–389, (Austin, TX, USA), November 1994.
15. H.-Y. Shum, K.-T. Ng, and S.-C. Chan, "Virtual reality using the concentric mosaic: Construction, rendering and data compression," in *Proceedings of the IEEE International Conference on Image Processing ICIP-2000*, **3**, pp. 644–647, (Vancouver, Canada), September 2000.

16. L. Luo, Y. Wu, J. Li, and Y.-Q. Zhang, "3-D wavelet compression and progressive inverse wavelet synthesis rendering of concentric mosaic," *IEEE Transactions on Image Processing* **11**, pp. 802–816, July 2002.
17. P. Eisert, E. Steinbach, and B. Girod, "Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views," *IEEE Transactions on Circuits and Systems for Video Technology* **10**, pp. 261–277, March 2000.
18. P. Ramanathan, M. Flierl, and B. Girod, "Multi-hypothesis disparity-compensated light field compression," in *Proceedings of the IEEE International Conference on Image Processing ICIP-2001*, **2**, pp. 101–104, (Thessaloniki, Greece), October 2001.
19. M. Held and R. M. Karp, "The traveling salesman problem and minimun spanning trees: Part ii, mathematical programming 1," 1971.
20. S. Timsjo, "An application of lagrangian relaxation to the traveling salesman problem," Tech. Rep. IMaTOM-1999-02, Malardalen University, Department of Mathematics and Physics, Sweden, June 1999.
21. W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM Journal on Mathematical Analysis* **29**(2), pp. 511–546, 1998.
22. J. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang, "Memory-constrained 3-D wavelet transform for video coding without boundary effects," *IEEE Transactions on Circuits and Systems for Video Technology* **12**, pp. 812–818, September 2002.
23. A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Commun. Pure Appl. Math.* **45**, pp. 485–560, 1992.
24. F. W. Wheeler and W. A. Pearlman, "Low-memory packetized SPIHT image compression," in *Conference Record of the Thirty-Third Asilomar Conference on Signals, Systems, and Computers.*, **2**, pp. 1193–1197, (Pacific Grove, CA, USA), October 1999.
25. K. K. Lin and R. M. Gray, "Video residual coding using SPIHT and dependent optimization," in *Proceedings of the Data Compression Conference 2001*, pp. 113–122, (Snowbird, UT, USA), March 2001.
26. S. Cho and W. Pearlman, "3-D wavelet coding of video with arbitrary regions of support," *IEEE Transactions on Circuits and Systems for Video Technology* **12**, pp. 157–171, March 2002.
27. S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding," *IEEE Transactions on Circuits and Systems for Video Technology* **10**, pp. 725–743, August 2000.
28. G. Minami, Z. Xiong, A. Wang, and S. Mehrotra, "3-D wavelet coding of video with arbitrary regions of support," *IEEE Transactions on Circuits and Systems for Video Technology* **11**, pp. 1063–1068, September 2001.
29. W.-C. Chen, J.-Y. Bouguet, M. H. Chu, and R. Grzeszczuk, "Light field mapping: Efficient representation and hardware rendering of surface light fields," in *Computer Graphics (Proceedings SIGGRAPH 02)*, pp. 447–456, July 2002.