Light Field Compression Using Disparity-Compensated Lifting and Shape Adaptation

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

***Abstract*—We propose *disparity-compensated lifting* for wavelet compression of light fields. With this approach, we obtain the benefits of wavelet coding, such as scalability in all dimensions, as well as superior compression performance. Additionally, the proposed approach solves the irreversibility limitations of previous light field wavelet coding approaches, using the lifting structure. Our scheme incorporates disparity compensation into the lifting structure for the transform across the views in the light field data set. Another transform is performed to exploit the coherence among neighboring pixels, followed by a modified SPIHT coder and rate-distortion optimized bitstream assembly. A view-sequencing algorithm is developed to organize the views for encoding. For light fields of an object, we propose to use shape adaptation to improve the compression efficiency and visual quality of the images. The necessary shape information is efficiently coded based on prediction from the existing geometry model. Experimental results show that the proposed scheme exhibits superior compression performance over existing light field compression techniques.**

***Index Terms*—Disparity compensation, lifting, light field, multiview image, shape adaptation, wavelet transform.**

I. Introduction

INTERACTIVE photorealistic three-dimensional (3-D) graphics has many potential applications ranging from scientific visualization and medical imaging to e-commerce and video games. Traditional rendering techniques for photo-realistic scenes, however, are typically computationally intensive. They additionally face the difficult problems of correctly modeling the surface properties, 3-D scene geometry, and lighting to obtain satisfactory results. Image-based rendering has emerged as an important 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. Since image-based rendering involves only resampling the acquired image data, it is particularly attractive for interactive applications.

A *light field*, described by Levoy and Hanrahan [1] and Gortler *et al.* [2], is a general image-based rendering data set. It captures the outgoing radiance from a particular scene or object, at all points in 3-D space and in all directions. In 3-D space, a light field depends on five independent variables, three for the viewing positions and two for the viewing angles. However, in free space without obstructions that block the radiance, the light field can be parameterized as a four-dimensional (4-D) data set since radiance is constant along lines in free space. In practice, the light field can be sampled and parameterized by capturing the scene or object with a two-dimensional (2-D) set of 2-D camera views. A novel view from an arbitrary position and direction can be generated by appropriately combining image pixels from the acquired views.

One of the main challenges with light fields is their enormous size. The uncompressed size for a large photo-realistic light field can easily exceed tens of Gigabytes [3]; therefore, compression is essential to any practical system.

A 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 viewpoints. In addition, it is desirable to have a scalable representation of the light field, which allows the system to efficiently adapt to varying storage capacities, transmission bandwidths, display devices, and computational 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]. In [4]–[6], the (DWT) has been proposed to exploit coherence as well as to achieve scalability. In [6], for instance, the multilevel 4-D Haar transform is directly applied to the 4-D light field data set, followed by a 4-D extension of the *set partitioning in hierarchical trees* (SPIHT) algorithm [7]. 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 low compression efficiency for both coders.

A common technique in light field compression to address the parallax problem is *disparity compensation*, akin to motion compensation in video coding. It is used in some prediction-based coders [8], [9], which have good compression efficiency but provide only limited support of scalability. In [10], disparity compensation is incorporated into a scalable coder by applying the 4-D wavelet coder described in [6] to an aligned reparametrization of the views based on an explicit geometry model. The problem with this scalable coder, however, is that the resampling process involved is irreversible and, therefore, introduces degradation in image quality.

Manuscript received October 4, 2003; revised January 26, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Trac D. Tran.

The authors are with the Information Systems Laboratory, Stanford University, Stanford, CA 94305 USA (e-mail: chuoling@stanford.edu; zhuxq@stanford.edu; pramanat@stanford.edu; bgirod@stanford.edu).

Digital Object Identifier 10.1109/TIP.2005.863954

1057-7149/$20.00 © 2006 IEEE

These effects mirrors recent work in 3-D subband coding of video signals. Many attempts have been made to incorporate motion compensation into the 3-D subband video coding framework [11]–[13]. Earlier works are somewhat unsatisfactory either for reasons similar to the resampling problem in light field compression, or because the displacement vector field is severely restricted. Recently, a technique called *motion-compensated lifting* [14]–[18] has been proposed, which successfully incorporates unrestricted motion compensation into 3-D subband coding in a reversible fashion.

For light field compression, we propose a wavelet coding scheme that achieves reversibility using a technique analogous to motion-compensated lifting for video. The proposed scheme, *disparity-compensated lifting*, incorporates disparity compensation into the DWT using the lifting structure. We extend the scheme to handle a more general representation of light fields, namely unstructured light fields [19], allowing arbitrary camera positions instead of restricting the views to a 2-D grid. For light fields describing an object with extraneous background, shape adaptation is further adopted to improve compression efficiency and reconstruction quality. Compared with existing techniques, the proposed scheme has several advantages. Reversibility resulting from the disparity-compensated lifting framework improves the compression performance. The wavelet transform allows scalability in different dimensions. The incorporation of shape-adaptation enhances compression efficiency as well as the visual quality of reconstructed views.

The remainder of the paper is organized as follows. In Section II, we provide a brief survey of light field compression using disparity compensation and discuss some open problems. The proposed approach is presented in Section III, where we describe in detail the major steps involved, including the inter- and intra-view transforms, coefficient coding and bitstream assembly, together with a discussion on scalability of the system. In Section IV, we introduce a method for sequencing the unstructured camera views of general light fields for applying the inter-view transform. In Section V, we present our shape adaptation techniques and describe a method for shape coding when the geometry model is approximate. The experimental results and comparisons with prior techniques are shown in Section VI.

II. Prior Work

Most light field compression approaches incorporate some form of disparity compensation for compression efficiency. Disparity compensation has originally been proposed for stereo and multiview image compression [20]–[24] and is also extensively used in compression of concentric mosaics, a 3-D data set for image-based rendering [25]–[30].

Existing light field compression approaches with disparity compensation generally fall into one of two categories: coders that use a prediction-based structure or coders that reparameterize the views onto the geometry.

To encode the views, the prediction-based coders apply disparity-compensated prediction from previously encoded views and encode the prediction residuals. In [31] and [32], Tong and Gray use VQ To encode intra frames and prediction residuals after disparity compensation, achieving reasonable compression efficiency and decoding speed. In [8], Magnor and Girod describe a disparity-compensated predictive method with a block-wise discrete cosine transform (DCT) and run-level-coding of the coefficients. In [33], Zhang and Li describe a similar disparity compensation method with multiple reference views. In [9], disparity values are inferred from an explicit geometry model. The scheme is extended in [34] to include disparity compensation from multiple reference views and multiple-hypothesis prediction, allowing multiple coding modes for each block in the view and performing mode selection using the Lagrangian multiplier technique. These coders exhibit good compression performance and viewpoint scalability; however, reconstruction quality and image resolution scalability are not supported.

In the second category of coders, an explicit geometry model is used to warp all the views to a common reference frame and resample the aligned views on a common grid. In [35]–[37], light fields are first warped onto the surface of a geometry model to form the so-called *surface light field*. Transform coding or matrix factorization techniques are then applied to facilitate compression as well as rendering. In [10], a texture map-based approach is proposed. An approximate geometry model is first estimated from the light field [38]. Using the geometry model, the various views in the data set are warped onto a global texture map reference frame, 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 acquired 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 warping the views is usually not reversible. Careful design of the warping functions can reduce these problems, but, nevertheless, the backward warping functions generally do not invert the forward warping functions. As a result, the reconstruction of the acquired views not only exhibits the quantization noise introduced by lossy compression of the wavelet coefficients, but also inevitably inherits the distortion arising from the mismatch between the forward and backward warping processes. Moreover, the approach can only encode the portions in each view covered by the geometry model because of the underlying parametrization. Note that these problems are not specific to the texture map wavelet coding approach described above, they arise for all light field compression and rendering systems involving a similar resampling process of the acquired views [1], [2], [4], [35]–[37]. The quality degradation due to resampling can affect the quality of the rendered view [19]. Accordingly, for light field compression, our objective is to minimize the distortion in the reconstruction of the acquired views for a given bit-rate constraint.

In the Section III, we describe a novel approach to solve the problems caused by the resampling process. Disparity compensation is effectively incorporated into the inter-view transform

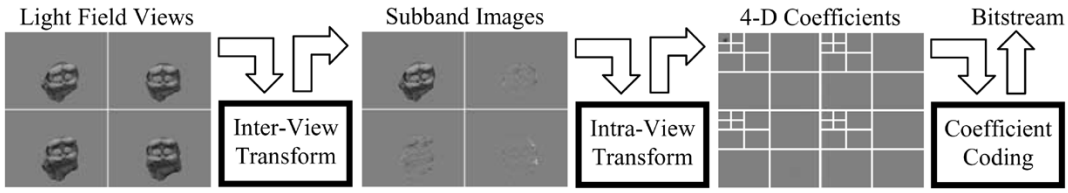

Fig. 1. System architecture: The input is a simplified light field containing only four views, which is represented, as a special case, as a 2-D array of views. The inter-view transform is first applied, both horizontally and vertically. The resulting subband images, including the LL, LH, HL, and HH subbands, are further decomposed by the 2-D intra-view transform. Finally, the wavelet coefficients are encoded to generate the compressed bitstream.

so that reconstruction quality of the acquired views are only affected by coefficient quantization. Additionally, in contrast to prediction-based coders, scalability is naturally supported by the DWT in the proposed scheme.

III. Light Field Compression Using Disparity-Compensated Lifting

In the proposed light field compression scheme, a wavelet transform is performed on all dimensions of the light field data set, through an inter-view transform and an intra-view transform. The inter-view transform is carried out using disparity-compensated lifting. The resulting subband images from the inter-view transform still exhibit coherence among neighboring pixels. Therefore, the 2-D image transform, i.e., the intra-view transform, is performed on each subband image to exploit the remaining coherence. To encode the final DWT coefficients, a modified version of the SPIHT algorithm is applied in 2-D for each of the transformed subband images.

The basic architecture of the proposed system is shown in Fig. 1, for the special case where the light field data set is represented as a 2-D array of views. In such a case, the inter-view transform is carried out by applying one-dimensional (1-D) transforms horizontally and vertically across the 2-D array of views. To handle general light fields with arbitrary camera settings, a view-sequencing method is developed to organize the data set as a sequence of views so that the inter-view transform can be applied efficiently.

In the following subsections, each component of the system, including the inter-view transform, intra-view transform, coefficient coding and bitstream assembly will be described. In addition, the scalability provided by the proposed system will be discussed. View sequencing will be presented in Section IV.

A. Inter-View Transform Using Disparity-Compensated Lifting

Lifting is a procedure that can be used to implement DWTs [39]. It is shown that any two-band subband transform with FIR filters can be implemented using the lifting structure [40]. Suppose that, in the context of light field compression, we have a sequence of N views, $x[n], n = 0, \ldots, N-1$. The order of the view sequence can be obtained either by the natural structure of the data set or by applying the view-sequencing algorithm that will be discussed in Section IV. Assuming N is even for simplicity, we split up this sequence into two sets of $N/2$ views: an even set $x_0[k], k = 0, \ldots, N/2-1$, and an odd set $x_1[k], k = 0, \ldots, N/2-1$. Wavelet analysis can be factored into one or more lifting steps, each consisting of a prediction and an update filter. The lifting procedure 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 nonlinear or data-adaptive filters, while still preserving reversibility.

For light field compression, we incorporate disparity compensation into the prediction and update filters. In particular, we use an explicit geometry model, estimated from the light field data set [38], for disparity compensation. Note that in video coding with motion-compensated lifting, multiple sets of motion vectors are often needed, including the one set between two neighboring video frames for the prediction filter, possibly a different set in the reverse direction for the update filter, and other sets that span multiple frames. For the proposed light field compression scheme, the disparity values between any pair of views in the data set can be derived from the geometry model. This makes the geometry model a compact representation of the disparity values. Complicated issues in video coding such as motion estimation and motion vector coding are, thus, eliminated. In addition, compared to block matching in video coding, disparity compensation using the geometry model generally provides better prediction of the signal. The resulting prediction residual in the high-pass subband images is smooth and free of blocking artifacts, suitable for the subsequent intra-view wavelet transform and coefficient coding stages for the purpose of compression.

We can formulate disparity compensation as a function that warps a view from one viewpoint to another. Let $v_0[k]$ and $v_1[k]$ denote the viewpoint, 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 viewpoint $v_0[k]$ 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 viewpoint $v_1[k]$ to $v_0[k]$. As an example, $w_{01}^{(k)}(x_0[k])$ denotes the warped view (with viewpoint $v_1[k]$), derived from the given view $x_0[k]$ (with viewpoint $v_0[k]$).

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 onto on the image plane at $v_0[k]$ to yield location p_0. The pixel value at p_0 is extracted from $x_0[k]$ using bilinear interpolation and assigned to p_1 as $w_{01}^{(k)}(x_0[k])$.

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:

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

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

$$= (x_0[k] - \frac{1}{2}w_{10}^{(k)}(w_{01}^{(k)}(x_0[k]))) + \frac{1}{2}w_{10}^{(k)}(x_1[k])$$

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

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

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 $x_1[k]$. The scaling factors needed to normalize the transform are omitted in (1). Note that the lifting structure is not limited to the Haar wavelet. Any DWT can be factorized into lifting steps [40].

Disparity-compensated lifting effectively incorporates disparity compensation into the DWT while maintaining the reversibility of the transform. In addition, the lifting structure also allows in-place calculation of the wavelet transform, i.e., the original samples $x_0[k]$ and $x_1[k]$ can be overwritten by the subbands $y_0[k]$ and $y_1[k]$ without having to allocate new memory [39]. Moreover, a memory-efficient implementation using a pipeline structure has also been proposed which is especially suitable for interactive rendering applications [30], [41].

1) 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 [42] are adopted because of their simplicity and effectiveness. Typically, the $5/3$ wavelet, due to its symmetric support which corresponds to bidirectional prediction and update, yields better performance than the Haar wavelet, at the cost of increased computation. To reduce computation, 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.

2) Multilevel 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 viewpoints $v_0[k]$. If the number of viewpoints is sufficiently large, a multilevel transform can be performed. The inter-view transform can be applied again, on $y_0[k]$, with $v_0[k]$ as the corresponding viewpoints. This procedure can be repeated several times depending on the density and total number of the viewpoints.

By applying multiple levels of the transform, despite that we are using wavelet kernels with relatively short supports such as the Haar and the 5/3 wavelet, the coherence in a larger neighborhood is in effect being exploited. In addition, the sequence $y_0[k]$ can be re-ordered before applying the inter-view transform to exploit the coherence among views more efficiently, as described in Section IV.

B. Intra-View Transform

After the inter-view transform, there is remaining coherence among neighboring pixels within each resulting subband image, especially for the low-pass subbands. To further exploit this, the intra-view transform is applied to each subband image using a multilevel 2-D DWT. The biorthogonal Cohen–Daubechies–Feauveau 9/7 wavelet [42], popular for image compression, is chosen for the intra-view transform. Note that due to the irrational coefficients of the 9/7 wavelet, the intra-view transform is in general not reversible. Nevertheless, for lossy compression this irreversibility in the intra-view transform has only minimal impact on compression efficiency. An extension to lossless compression should include an integer-to-integer intra-view transform, such as defined in the JPEG2000 standard [43], together with rounding of the disparity-compensated signal that constitutes an integer-to-integer inter-view transform. However, the main scope of this work is lossy compression and the lossless extension is not incorporated in our current implementation.

C. Coefficient Coding

To encode the DWT coefficients, the SPIHT algorithm is chosen for its computational simplicity and high compression efficiency [7]. It is applied to each subband image separately and modified to operate in a block-wise manner.

1) Two-Dimensional Coding Versus Higher-Dimensional Coding: It may seem natural to use a 3-D or 4-D SPIHT coder to encode the inter-view and intra-view transformed coefficients as in [6] and [10]. Our experiments, however, show that compression performance of the higher-dimensional SPIHT coder is inferior to that of its 2-D counterpart for the proposed system. The main reason why 2-D coding is better is that the subband images maintain their own viewpoints after disparity-compensated lifting, unlike the aligned structure in [6] and [10]; therefore, coefficients at the same position in different subband images no longer necessarily correspond to the same point in the original scene. Hence, the assumption of high correlation across the views for higher-dimensional SPIHT coding no longer holds.

Aside from the above observation, higher-dimensional SPIHT coding is more computationally demanding and less flexible for bitstream truncation. Two-dimensional SPIHT coding is, therefore, chosen in our system, with each subband image encoded separately.

2) Block-Wise Spiht: For better exploitation of the local statistics as well as memory efficiency, we further modify the SPIHT coder to regroup the DWT coefficients in each subband image into individual blocks and encode them separately. Similar ideas have been proposed for image compression [44] and video residual image coding [45].

For each subband image, the coefficients are re-grouped into blocks as illustrated in Fig. 2. The SPIHT algorithm is then separately applied to each coefficient block, generating a corresponding bitstream. 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

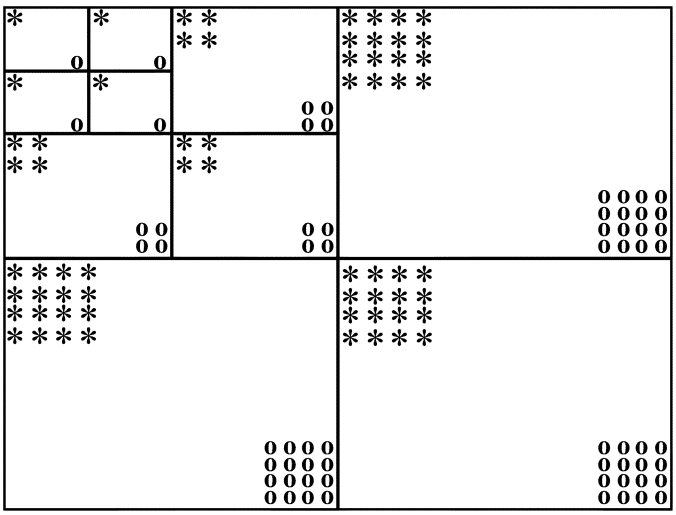

Fig. 2. Block-wise SPIHT: The transform coefficients are divided into blocks, each containing spatially neighboring coefficients from all intra-view subbands. For example, the coefficients at the locations labeled by $*$ are grouped into one block, and those labeled by o are grouped into another block.

globally for the entire image. The block-wise SPIHT coder also lowers the memory requirement [44] and allows the ability to randomly access part of an image without having to decode its entirety, an important feature for light field rendering. Note that although the coefficients are coded together within each block, the intra-view wavelet transform is performed on the entire image.

D. Scalability

For a given bitstream, we can decode only part of it to reduce the bit rate at the cost of increased distortion, yielding reconstruction-quality scalability.

Viewpoint 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 has to be modified as described in [46]. Note that this modification has not been incorporated into our current implementation.

IV. View Sequencing

In order to apply the inter-view transform, the camera views in the data set must be organized as a sequence of views, with a specific scanning order for compression. For the light fields acquired by cameras positioned as a 2-D array encompassing the scene, the data sets can be easily represented as a 2-D array of camera views [1], [47]. 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 [19]. 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 [19] where the cameras are not positioned on a regular grid, an effective view sequencing method should arrange the camera views such that neighboring views in the sequence exhibit high coherence for the wavelet transform to decorrelate the signals more effectively.

For a light field data set, the coherence between two camera views can be accounted for, for instance, by the angular difference of the viewing directions and the difference of the distances from the camera centers to the scene [19]. In the data sets we work on, cameras are densely positioned surrounding the scene, and they are approximately equally distant to the center of the scene and looking at it. Hence, only the angular difference in the viewing directions is of concern, which roughly corresponds to the distance between each pair of cameras. In other words, as an approximation applied for the data sets of interest, camera views that are taken from nearby positions are assumed to exhibit higher coherence. The view sequencing method should attempt to minimize the camera distance between neighbors in the resulting view sequence.

As a result, we propose to formulate the view sequencing problem as the *travelling salesman problem (TSP)* [48], 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. Each node corresponds to a view, and the cost of a path connecting two views is defined as the Euclidian distance between their corresponding camera centers. Note that, for general data sets, other metrics, such as the camera-to-scene difference, can be incorporated into the cost calculation as the penalty measurement defined in [19]. The fact that the path returns to the initial node also facilitates periodic extensions for the inter-view transform at the boundary of the view sequence.

Note that the optimal solution of TSP, which is NP-complete, does not necessarily guarantee the best compression performance since the problem formulation is based on several approximations. A suboptimal solution suffices for the purpose of systematically arranging the data sets into a view sequence so that the inter-view transform can be carried out efficiently. Since the Euclidian distance is a symmetric metric, we adopt algorithms proposed for symmetric TSP based on Lagrangian relaxation [48], [49].

We further propose to group the camera positions in a data set into several clusters using, for instance, the K-means clustering algorithm, and then independently sequence the views within each cluster. For each cluster, TSP considers only a fraction of the views in the data set, and, hence, the complexity is significantly reduced. Additionally, as a result, the views in one cluster are encoded independently from those in other clusters. This is beneficial when the user only navigates part of the scene rather

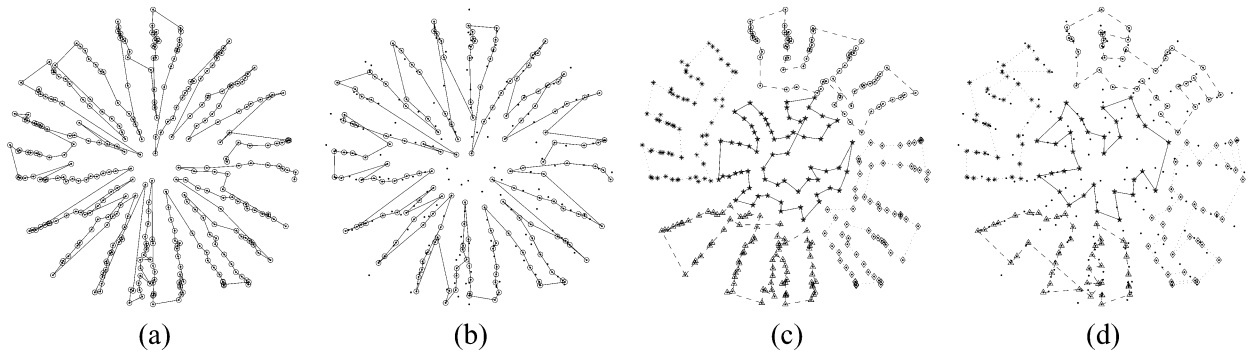

Fig. 3. View sequencing for *Bust*: The cameras are approximately distributed on a hemisphere surrounding the object. The dots denote the camera positions viewing from the top of the hemisphere. The lines denote the view sequence for different levels of the inter-view transform. (a) Heuristic, one-level. (b) Heuristic, two-level. (c) TSP, five clusters, one-level. (d) TSP, five clusters, two-level.

than the whole data set, since only the data in the clusters being navigated need to be accessed.

As an example of the unstructured light fields, the camera positions of the *Bust* data set are shown in Fig. 3. The cameras are distributed approximately on a hemisphere surrounding the target object. The result of a heuristic view sequencing method that approximately traverses along the longitude lines of the hemisphere is shown in Fig. 3(a). For view sequencing of the second level of the inter-view transform, the heuristic method simply follows the even samples in the current view sequence, i.e., the camera positions corresponding to the low-pass subband images, as what is typically done for data sets with a regular grid, as shown in Fig. 3(b).

The results using the proposed TSP method with five clusters are shown in Fig. 3(c). For the second level, instead of following the direction of the current view sequence as in the heuristic approach, TSP is applied again to the set of the even samples in order to generate a sequence that can possibly exploit the coherence in a different direction, as shown in Fig. 3(d). The compression performance using the proposed TSP method compared to the heuristic method will be discussed in Section VI-C.

V. Shape Adaptation

When the light field of interest is an object, the constituent images contain extraneous background pixels and discontinuities at the object boundaries. In the former, we encode unnecessary pixels. In the latter, there is increased energy in the high-frequency components. In both cases, this leads to inefficiency in the coding. We, therefore, propose to mitigate these two effects by utilizing 2-D shape of the object in each view, obtained by image segmentation, when coding the light field data set.

In the proposed light field compression scheme, a geometry model of the scene is available at the decoder to provide the disparity values as discussed in Section III-A. If the geometry model is accurate such that projection of the geometry is consistent with the 2-D object shape in each view, the geometry model itself can account for the shape information. On the other hand, if the geometry model is just an approximation, it provides only approximate shape information. In this case, techniques for coding the exact shape, as will be discussed in Section V-B, are needed. An example of the exact shape and the approximate shape is illustrated in Fig. 4(a) and (b), respectively.

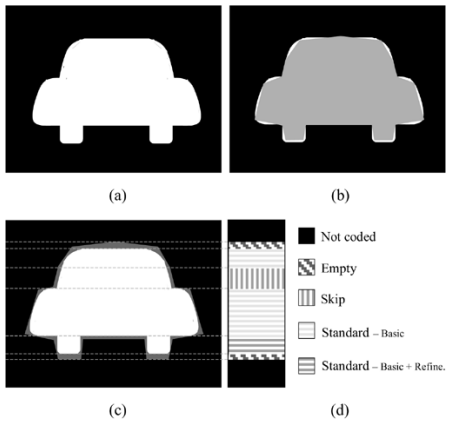

Fig. 4. Shape coding example. (a) Exact shape S showing a front view of a car. (b) Approximate shape $\hat{S}$ derived from a geometry model, in light gray and superimposed on S. (c) Dilated approximate shape $\hat{S}$, in dark gray with S superimposed on it. The dashed lines divide the view into regions with different mode selections. (d) Mode selection of each line is shown. For example, the lines in the region containing the tires are using *Standard* mode with both *Basic* and *Refinement* passes. One of the regions is using *Skip* mode because $\hat{S}$ is already identical to S in this region as can be seen in (b).

A. Shape-Adaptive Transform and Coefficient Coding

Utilizing 2-D object shape information is beneficial for all three stages of the proposed light field compression system: the inter-view transform, the intra-view transform, and coefficient coding.

In image and video compression, shape adaptation has been proposed for the image-domain wavelet transform and the wavelet coefficient coding methods [50], [51]. Similarly, for the intra-view transform stage in light field compression, we apply the shape-adaptive DWT (SA-DWT) [50] on each subband image, generating as many wavelet coefficients as the object pixels. It also avoids performing the transform across object boundaries, contributing to improved coding efficiency and enhanced reconstruction quality. For coefficient coding, the SPIHT algorithm is modified to disregard zero-trees that contain only background pixels as described in [51]. 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 [51]. Therefore, the need of arithmetic coding is eliminated, and

coding complexity can be reduced without sacrificing much compression efficiency.

Specifically for light field compression, we further incorporate the 2-D object shape, together with the geometry model, to exploit the inter-view coherence in the data set. Using the 2-D shape, disparity compensation at the object boundaries can be improved especially when the geometry model is only an approximation.

In particular, with an inaccurate geometry model, an object pixel in one view may be disparity-compensated to the background in another view. The contrast between object and background pixels may give rise to large residual errors that are expensive to code. With knowledge of exact object boundaries, on the other hand, the prediction can be obtained from the nearest object pixel instead of the background.

Another problem with disparity compensation from inaccurate geometry is that some object pixels near the boundary do not have a corresponding point on the geometry. In this case disparity compensation is not applicable, resulting in large residual errors at the edges. With the exact 2-D shape, all object pixels are easily identified. The disparity values of those pixels unaccounted for by the geometry can be simply extrapolated from neighboring pixels; thus, all the object pixels can be disparity-compensated properly.

In practice, an accurate geometry model is not always possible to acquire, whereas the 2-D object shape in each view can be conveniently obtained by image segmentation techniques. Using the 2-D object shape as an auxiliary to the approximate geometry model is, therefore, advantageous for light field compression. Furthermore, side information of the 2-D object shape can be coded, in return, with the assistance of the approximate geometry model as demonstrated in the next subsection.

B. Shape Coding

If the geometry model is just an approximation, we propose to code the exact shape, denoted by S, using the available approximate shape derived from the geometry, denoted by $\hat{S}$. $\hat{S}$ is dilated for several iterations until the white pixels in the dilated image cover all of the white pixels in S. We denote the dilated version of $\hat{S}$ by $\tilde{S}$, illustrated inFig. 4(c), which is now used to predict S. The number of dilations is transmitted, and the decoder, knowing this number, can recover $\tilde{S}$ since the geometry is also available.

$\tilde{S}$, $\hat{S}$, and S are compared for the horizontal scan lines containing white pixels in $\tilde{S}$. We denote a line in $\tilde{S}$ by $\tilde{L}$, the corresponding line in $\hat{S}$ by $\hat{L}$ and that in S by L. Three modes are defined for all possible situations.

- *Empty* mode: This mode is selected when L contains no white pixels, i.e., the object is not contained in this line.
- *Skip* mode: This mode is selected when L contains white pixels and it is identical to $\hat{L}$, i.e., the approximate shape is already identical to the exact shape.
- *Standard* mode: This mode is selected when L contains white pixels but is not identical to $\hat{L}$. In this case, $\tilde{L}$ is used to predict L. Two passes are further defined for this mode.
  - *Basic* pass: The distance between the left-most white pixel in L and that in $\tilde{L}$ is recorded, followed by the distance between the right-most white pixel in L and that in $\tilde{L}$. The two distances are predictively encoded using the corresponding distance from the line above.
  - *Refinement* pass: If the line contains more than one run of white pixels in L, it is signaled in the refinement pass. Starting from the left-most white pixel, the length of each run in L is recorded, alternating between white runs and black runs. Note that the length of the last white run does not need to be encoded since it can be derived from the *Basic* pass.

The mode selection of each line for the given example is shown in Fig. 4(d). The mode selections and the associated distances and run-lengths are combined to form a set of symbols, which are further encoded using an adaptive arithmetic coder. Note that as a more accurate geometry is used, the *Skip* mode is selected more often; hence, lower bit rate is needed for shape coding.

VI. Experimental Results

Experimental results are shown for two types of light field data sets: arranged in an array, and unstructured. The first type consists of two data sets: *Garfield* and *Penguin*. Examples of these data sets are shown in Figs. 11 and 12. Each data set has a hemispherical view arrangement, where there are eight latitudes each containing 32 views, each view with a resolution of 288×384 pixels. The resulting data set can be directly parameterized as a 2-D (8×32) array of 2-D (288×384) views. For each data set, an approximate geometry model with 2048 triangles is reconstructed from the views, using the method described in [38]. The object shape for each view is obtained by image segmentation using a simple thresholding procedure. Using the proposed shape coding method with the assistance of the geometry model, the shape information is encoded at around 0.008 bpp for both data sets, compared to the 0.017 bpp by directly applying *JBIG* [52] on the exact shape S and around 0.023 bpp by applying *JBIG* on the difference between S and the approximate shape $\hat{S}$. The difference image is more difficult to encode with JBIG since it typically has more transitions than the shape itself. The proposed shape coding method overcomes this issue by employing a run-length-coding-like approach as described in Section V-B. The overhead for shape coding is included. The four-level intra-view transform with the 9/7 wavelet is chosen for these two data sets as it gives the best empirical performance.

The second type of data sets also includes two data sets: *Bust* and *Buddha*. Examples of the data sets are shown in Figs. 13 and 14. *Bust* consists of 339 views of a real-world object; each view has a resolution of 480×768. The geometry model and the camera parameters are estimated from the acquired camera views using methods described in [37]. *Buddha* is a computer synthesized data set with 281 views, each with a resolution of 512×512, together with a known geometry model and camera parameters. In both data sets, the camera positions are approximately distributed on a hemisphere surrounding the object. The camera positions in *Bust* and the results of view sequencing (Section IV) are shown in Fig. 3. For these two data sets, unlike *Garfield* and *Penguin*, the geometry models are accurate. Therefore, 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. The five-level intra-view transform with the 9/7 wavelet is used for these two data sets.

For light field rendering, a novel view is in general rendered by appropriately combining image pixels in the corresponding reference views, i.e., a number of camera views with viewpoints closest to this particular novel viewpoint, possibly via the geometry model [1], [19]. Therefore, during a rendering session, only the camera views that serve as reference views of the desired novel viewpoints have to be reconstructed from the coded data set. To evaluate the compression performance, in the following experiments, we assume that the user navigates throughout the entire scene such that every camera view needs to be reconstructed. As a result, we consider the bit rate and the reconstruction quality for all camera views in the data set.

For bit allocation among coefficient blocks, we use Lagrangian multiplier techniques similar to those in [53] for image compression to choose the optimal truncation points for each bitstream (coefficient block) so as to maximize the reconstruction quality for all camera views in the data set subject to different bit-rate constraints.

Note that we can safely assume the distortion (mean squared error) in a reconstructed camera view is proportional to that in the wavelet coefficients, and warping a view to a different viewpoint preserves the distortion originally in the view. If we further assume that the reconstruction error in one view is uncorrelated to that in any other view, the distortion between the novel view rendered from the reconstructed camera views and that rendered from the original camera views can be approximated by a weighted sum of the reconstruction distortion in each of its reference views [54]. As a result, the reconstruction quality of the camera views provides a direct indication of the quality of the rendered views.

Shape adaptation is incorporated in the experiments unless otherwise mentioned. With shape adaptation, 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 in bits 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 considering only the object pixels in all camera views. The compression performance is shown using rate-PSNR curves, which express the relation between the bit rate (bpop) and reconstruction quality in PSNR (in decibels). The geometry model is encoded at 0.027 bpop for *Garfield*, 0.034 bpop for *Penguin*, 0.018 bpop for *Bust*, and 0.013 bpop for *Buddha*. Specifically, each dimension of the 3-D coordinate of a geometry vertex is quantized and encoded by 21 bits, and the index of each vertex is represented by 10–12 bits. The camera parameters (camera position and viewing direction in each view) are encoded at around 0.008 bpop for *Garfield*, 0.010 bpop for *Penguin*, 0.002 bpop for *Bust*, and 0.001 bpop for *Buddha*. The geometry model and the camera parameters are, in general, used for rendering as well. Therefore, the bit rate for encoding such information is not included in the following rate-PSNR curves.

A. Inter-View Wavelet Kernels

The compression performance of four different inter-view wavelet kernels, Haar, $5/3$, truncated-Haar, and truncated-$5/3$, is compared along with the intra-coding scheme without inter-view transform. The rate-PSNR curves for *Garfield* are shown in Fig. 5. The one-level inter-view transform, i.e., inter-view transform both vertically and horizontally since the data set is represented as a 2-D array of views, is used. Only the luminance component is coded. Results for the other data sets, which are not shown, are similar [47], [55].

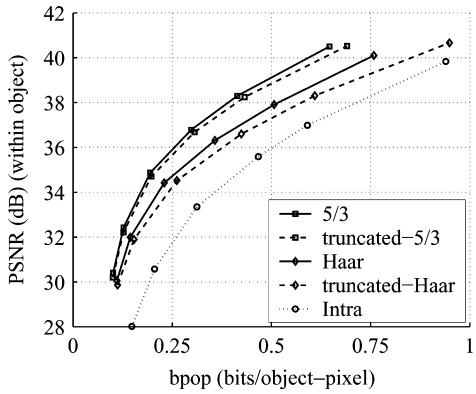

Fig. 5. Rate-PSNR curves for *Garfield* with different inter-view wavelet kernels, using the one-level inter-view transform.

At the same bit rate, the 5/3 wavelet performs about 1–1.5 dB better than the Haar wavelet. Compared to the intra-coding scheme, the inter-view transform with the 5/3 wavelet provides 3–4 dB gain in terms of PSNR at the same bit rate, or equivalently a bit rate reduction of 50% for the same reconstruction quality.

The truncated kernels perform worse than their nontruncated counterparts. Note that for video compression, cases have been reported where the nontruncated kernels give inferior performance than the truncated ones [16]. This is mostly due to the ghosting artifacts in the low-pass subband images, resulting from the occasional failure of motion compensation in the update lifting step, that are costly to encode. In the proposed light field compression scheme, the 3-D geometry model typically provides satisfactory results of disparity compensation; hence, the advantage of the truncated kernels is negligible. Instead, inferior compression performance due to aliasing in the low-pass subbands, as well as quality fluctuation in the reconstructed views, both result from absence of the update lifting step [56], [57], actually make the truncated kernels undesirable in spite of the reduced computational complexity.

B. Shape-Adaptation

To investigate the gain by shape adaptation, we conduct experiments with shape adaptation switched off for comparison. The $5/3$ wavelet with the one-level inter-view transform is used, and only the luminance component is coded. The rate-PSNR curves for *Garfield* and *Penguin* are shown in Fig. 6. Note that the measurement for bit rate is now bpp instead of bpop, since no shape information is available for the scheme without shape adaptation. For the same reason, the PSNR value is computed by averaging over all luminance pixels in the data set. The ratio between the number of object pixels and total number of pixels is 16.70% in *Garfield* and 13.13% in *Penguin*.

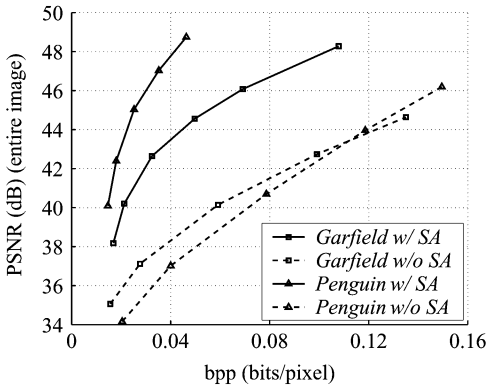

Fig. 6. Effect of shape adaptation: For the same reconstruction quality, shape adaptation reduces 60%–80% of the bit rate.

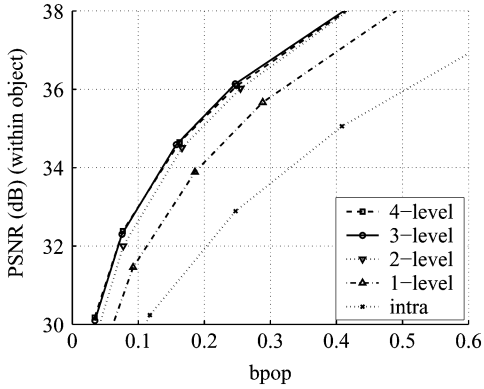

Fig. 8. Rate-PSNR curves for *Bust* with different inter-view transform levels, using the 5/3 wavelet.

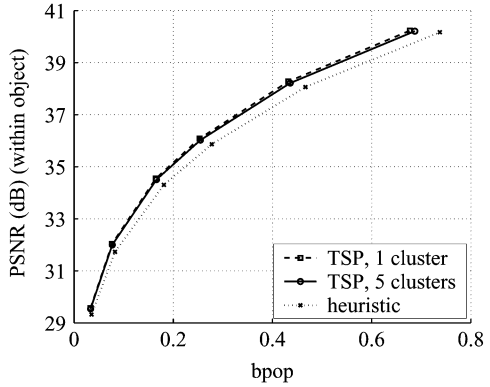

Fig. 7. Rate-PSNR curves for *Bust* with different view-sequencing methods, using the two-level inter-view transform.

Without shape adaptation, the block-wise SPIHT coder allocates no bits to encode the coefficient blocks having all zeros that typically happen at background blocks. Even so, the overhead for control information of those coefficient blocks still cannot be omitted. More importantly, the boundary regions containing both background and object pixels cannot be handled easily. On the opposite, shape adaptation reduces the bit rate by 60%–80% for the same reconstruction quality for both data sets due to omitting the overhead for all-background blocks and avoiding high-frequency components at the object boundaries, as discussed in Section V-A.

C. View Sequencing

The compression performance of the heuristic view-sequencing method and the TSP view-sequencing method with one and five clusters (Section IV), all using the two-level inter-view transform, is compared in Fig. 7. Results for the *Buddha* data set, which are not shown here, are similar.

The rate-PSNR curves show that the TSP method outperforms the heuristic method with a gain of 0.25 dB at the same bit rate. For the TSP method, the multiple-cluster case that groups the views into five clusters introduces slight degradation of the compression performance compared to the case considering the whole data set as a single cluster. This is because coherence across clusters is not exploited in the former case. Nevertheless, the additional advantages of the multiple-cluster case such as reduced complexity for sequencing and more efficient data access, as discussed in Section IV, make it the favorable approach. In the following experimental results for *Bust* and *Buddha*, the views are sequenced using the TSP method with five clusters.

D. Multiple Inter-View Transform Levels

In these experiments, the $5/3$ wavelet is used and only the luminance component is coded. The results for *Bust* using 1–4 levels of inter-view transform are shown in Fig. 8, along with the intra-coding scheme. There is about a 2-dB gain by applying the one-level transform over the intra-coding scheme, and a 1-dB gain by further applying the two-level transform. The gain diminishes with more levels of inter-view transform, and the performance degrades when using the four-level transform. This may be due to the fact that neighboring views in the four-level transform are too far apart to allow an efficient decomposition. Results for other data sets show similar performance [47].

E. Comparison With Existing Techniques

For *Garfield* and *Penguin*, we compare the proposed coder with the shape-adaptive DCT (SA-DCT) coder proposed in [58] and the texture map coder described in [10]. The experiments here are performed on color images using the (Y, Cb, Cr) color representation, with chrominance components down-sampled by a factor of 2 in each image dimension. The rate-PSNR curves are shown in Fig. 9. The bit rate (bpop) is derived from dividing the total bitstream length of the three channels by the number of object pixels in the luminance component. The reconstruction quality (PSNR) for the curves is computed using the luminance component only. Examples of the luminance component of the reconstructed views using the three coders at similar bit rates are shown in Figs. 11 and 12.

For the proposed coder, we use the four-level intra-view transform for the luminance, and the three-level intra-view transform for the sub-sampled chrominance components. The one-level $5/3$ wavelet is used for the inter-view transform. The compression procedure is applied to the three color channels separately using the same Lagrangian multiplier λ.

For the SA-DCT coder [58], the 2-D object shape is obtained and coded in the same way as the proposed coder. It uses a hierarchical prediction structure to encode all the views. Each

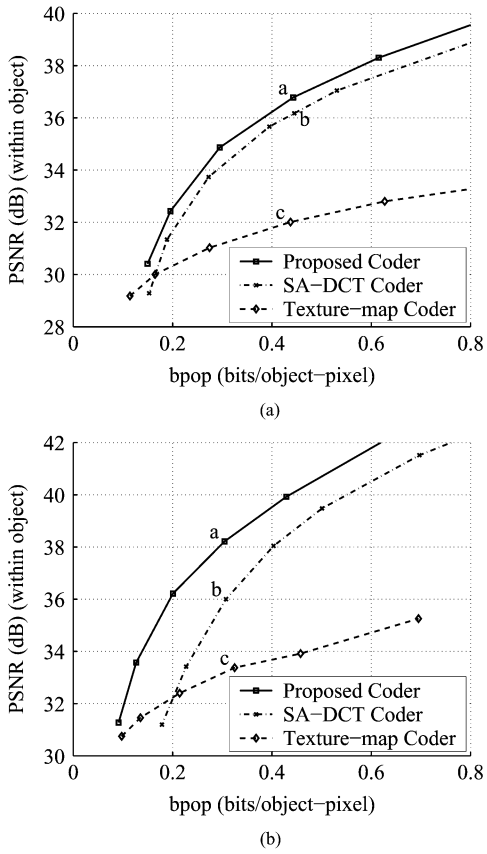

Fig. 9. Comparison of the SA-DCT coder, the texture map coder, and the proposed coder for *Garfield* and *Penguin*.

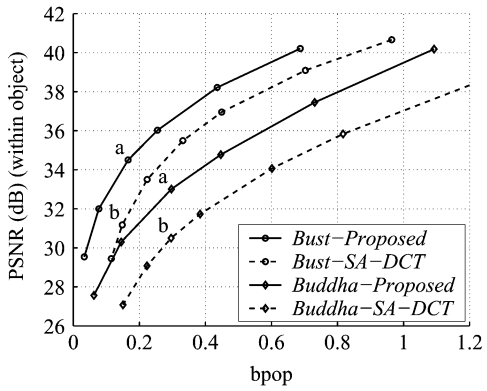

Fig. 10. Comparison of the SA-DCT coder and the proposed coder for *Bust* and *Buddha*.

view is divided into blocks of 8×8 pixels. For each block, there are three possible modes of coding: *INTRA*, image coding without prediction; *GEO*, disparity compensation from a geometry model followed by residual error coding; and *COPY*, copying from the block at the same position in a designated reference image. To encode each block with *INTRA* and *GEO* mode, an 8×8 SA-DCT [59], [60] is applied, followed by quantization and run-level-coding of the coefficients. Mode selection is based on minimizing rate-distortion Lagrangian cost function for each block. Each color channel is coded separately using the same quantization step size.

The texture map coder [10] is described in Section II. Bit allocation among channels is achieved by decoding the bitstream of each channel up to the same length as for the proposed coder to obtain the same bit allocation for the two coders.

Note that all three coders need the geometry model as side information. However, only the proposed coder and the SA-DCT coder need the shape information, since the texture map coder implicitly uses the approximate shape derived from the geometry. Consequently, the PSNR and bit-rate values for the texture map coder is obtained by only considering the object pixels within the approximate shape, which are a different from those of the other two coders that come from the exact shape.

Compared to the SA-DCT coder, the proposed coder achieves superior compression performance. In addition, the reconstructed views of the proposed coder do not exhibit blocking artifacts, which can be observed with the SA-DCT coder as shown in the magnified image in Fig. 12(b). The proposed coder additionally provides scalability for reconstruction quality, which is not supported in the SA-DCT coder.

Compared to the texture map coder, the compression performance of the proposed coder is consistently better, except for the very-low-bit-rate region, where the extra overhead for the shape information is no longer negligible. If the proposed coder only uses approximate shape from the geometry for fair comparison; however, it always performs better than the texture map coder. With increasing bit rate, the performance gap grows indefinitely since reconstruction quality of the texture map coder is limited to about 34 dB for *Garfield* and 36 dB for *Penguin* by the irreversible resampling process, whereas that of the proposed coder increases with bit rate. The proposed coder gains more than 6 dB in PSNR over the texture map approach for the high bit-rate regime in the plot. Alternatively, there is a reduction of 70% in bit rate for the same reconstruction quality. Furthermore, as shown in Figs. 11 and 12, the object shape is distorted for the texture map coder, whereas the proposed coder retains the original object shape.

For *Buddha* and *Bust*, we compare the proposed coder with the SA-DCT coder [58]. The rate-PSNR curves are shown in Fig. 10. Only the luminance component is coded. Examples of the reconstructed views using the two coders at similar bit rates are shown in Figs. 13 and 14. For the proposed coder, we use the two-level inter-view transform with the $5/3$ wavelet, along with the five-level intra-view transform, for both data sets. Note that the compression efficiency of the SA-DCT coder largely relies on the 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. More specifically, the two-level inter-view transform corresponds to $1/4$ of the views being intra-coded.

From the rate-PSNR curves in Fig. 10, we can see that the proposed scheme exhibits superior compression efficiency over the SA-DCT coder. For *Bust*, the proposed scheme outperforms the SA-DCT coder by a gain of 1.5–2 dB in terms of object PSNR, or equivalently a reduction of 20% in bit rate. In the case of *Buddha*, the gain in object PSNR for the proposed coder is around 2 dB, or equivalently the bit rate reduction is up to 30%. The proposed coder further achieves better visual quality in its

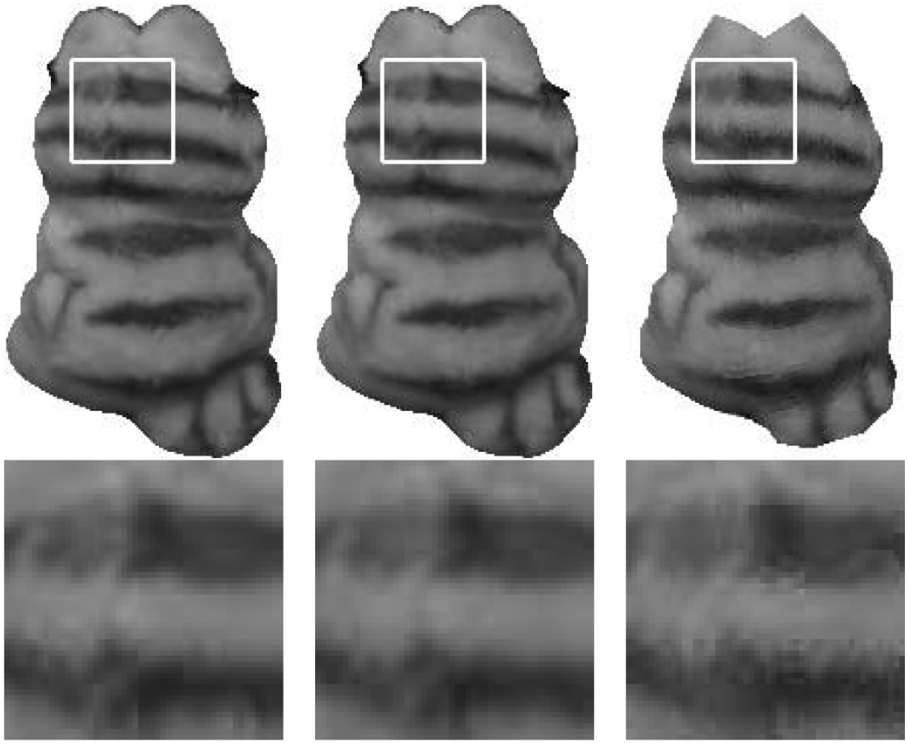

Fig. 11. Luminance component of *Garfield*: The reconstructed view from different coders, corresponding to the labeled points on the rate-PSNR curves in Fig. 9(a) are shown on the top row, with the white box labeling the area magnified on the bottom row. (a) Proposed at 0.443 bpop. (b) SA-DCT at 0.445 bpop. (c) Texture map at 0.437 bpop (bit rates include all color components).

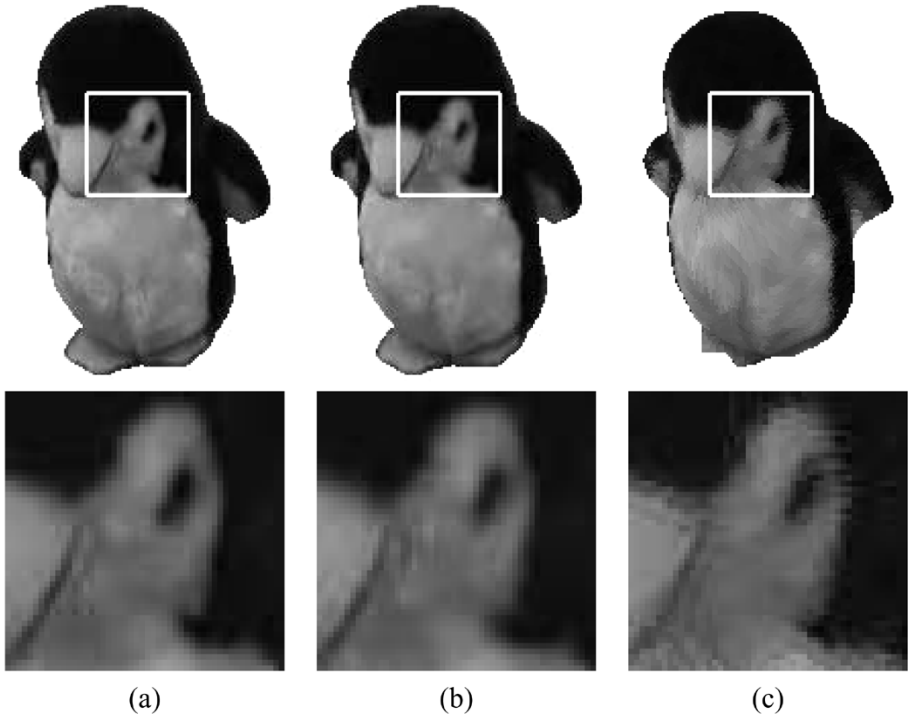

Fig. 12. Luminance component of *Penguin*: The reconstructed view from different coders, corresponding to the labeled points on the rate-PSNR curves in Fig. 9(b), are shown on the top row, with the white box labeling the area magnified on the bottom row. (a) Proposed at 0.305 bpop. (b) SA-DCT at 0.307 bpop. (c) Texture map at 0.325 bpop (bit rates include all color components).

reconstructions. As shown in Figs. 13(a) and 14(a), there are no blocking artifacts in the reconstructed views, which are observable in the magnified images of the corresponding SA-DCT coder results.

VII. Conclusion

We propose a novel approach for light field compression that uses disparity-compensated lifting for inter-view wavelet

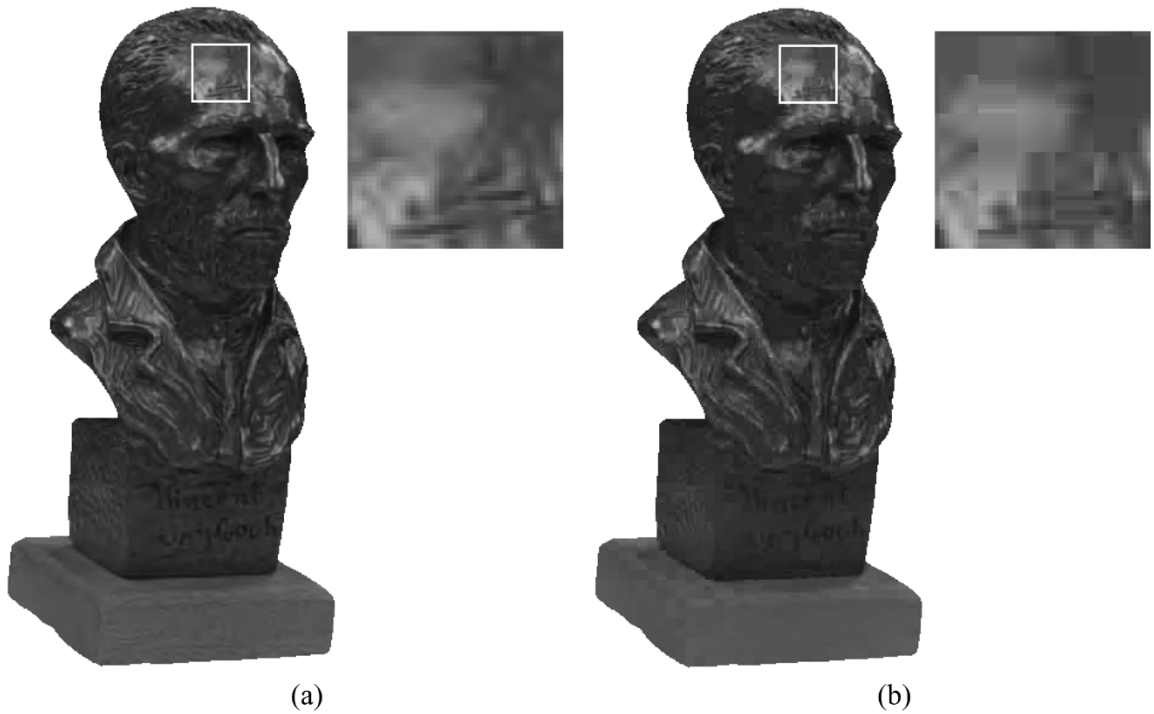

Fig. 13. Luminance component of *Bust*: The reconstructed view from different coders, corresponding to the labeled points on the rate-PSNR curves in Fig. 10 are shown with the white box labeling the area magnified beside. (a) Proposed at 0.166 bpop. (b) SA-DCT at 0.148 bpop.

Fig. 14. Luminance component of *Buddha*: The reconstructed view from different coders, corresponding to the labeled points on the rate-PSNR curves in Fig. 10, are shown with the white box labeling the area magnified beside. (a) Proposed at 0.297 bpop. (b) SA-DCT at 0.296 bpop.

coding. The lifting structure integrates disparity compensation into wavelet coding and remains reversible. Reconstruction of the acquired views is free of the distortion caused by the irreversible resampling process. The proposed scheme also supports scalability in image resolution, viewpoint, and reconstruction quality.

The scheme is extended to accommodate unstructured light fields by formulating the view sequencing problem as the TSP.

For light fields of an object with extraneous background, we propose to use shape adaptation techniques, which improves coding efficiency as well as visual quality of the reconstructed views. An efficient shape coding method is described for cases where the geometry model is approximate.

Experimental results from different types of light field data sets show that the compression efficiency of the proposed approach outperforms current state-of-the-art techniques. Compared with the texture map coder, a gain of more than 6 dB in overall reconstruction quality at the same bit rate is observed, or equivalently, a bit rate reduction of up to 70% at the same reconstruction quality. 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.

ACKNOWLEDGMENT

The authors would like to thank the University of Erlangen-Nuremberg for the *Garfield* and *Penguin* data sets used in this work and Intel Research for the *Buddha* and *Bust* data sets used in this work.

REFERENCES

[1] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. SIGGRAPH*, Aug. 1996, pp. 31–42.

[2] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. SIGGRAPH*, Aug. 1996, pp. 43–54.

[3] M. Levoy *et al.*, "The digital michelangelo project: 3D scanning of large statues," in *Proc. SIGGRAPH*, Aug. 2000, pp. 131–144.

[4] I. Ihm, S. Park, and R. K. Lee, "Rendering of spherical light fields," in *Proc. Pacific Graphics*, Seoul, Korea, Oct. 1997, pp. 57–68.

[5] I. Peter and W. Strasser, "The wavelet stream: Progressive transmission of compressed light field data," in *Proc. IEEE Visualization Late Breaking Hot Topics*, Oct. 1999, pp. 69–72.

[6] M. Magnor, A. Endmann, and B. Girod, "Progressive compression and rendering of light fields," in *Proc. Vision, Modeling, Visualization*, Erlangen, Germany, Nov. 2000, pp. 199–203.

[7] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, Jun. 1996.

[8] M. Magnor and B. Girod, "Data compression for light field rendering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 3, pp. 338–343, Apr. 2000.

[9] M. Magnor, P. Eisert, and B. Girod, "Model-aided coding of multi-viewpoint image data," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Vancouver, BC, Canada, Sep. 2000, pp. 919–922.

[10] M. Magnor and B. Girod, "Model-based coding of multi-viewpoint imagery," in *Proc. SPIE Visual Communications Image Processing*, vol. 1, Perth, Australia, Jun. 2000, pp. 14–22.

[11] D. Taubman and A. Zakhor, "Multi-rate 3-D subband coding of video," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 572–588, Sep. 1994.

[12] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 559–571, Sep. 1994.

[13] S. Choi and J. Woods, "Motion compensated 3-D subband coding of video," *IEEE Trans. Image Process.*, vol. 8, no. 2, pp. 155–167, Feb. 1999.

[14] A. Secker and D. Taubman, "Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Thessaloniki, Greece, Oct. 2001, pp. 1029–1032.

[15] B. Pesquet-Popescu and V. Bottreau, "Three dimensional lifting schemes for motion compensated video compression," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 3, Salt Lake City, UT, May 2001, pp. 1793–1796.

[16] L. Luo *et al.*, "Motion compensated lifting wavelet and its application in video coding," in *Proc. IEEE Int. Conf. Multimedia Expo*, Tokyo, Japan, Aug. 2001, pp. 481–484.

[17] A. Secker and D. Taubman, "Highly scalable video compression using a lifting-based 3D wavelet transform with deformable mesh motion compensation," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Rochester, NY, Sep. 2002, pp. 749–752.

[18] D. Taubman, "Remote browsing of JPEG2000 images," in *Proc. IEEE Int. Conf. Image Process.*, Rochester, NY, Sep. 2002, pp. 229–232.

[19] C. Buehler, M. Bosse, L. McMmillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Proc. SIGGRAPH*, Aug. 2001, pp. 425–432.

[20] M. Lukacs, "Predictive coding of multi-viewpoint image sets," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Tokyo, Japan, Mar. 1986, pp. 521–524.

[21] I. Dinstein, G. Guy, and J. Rabany, "On the compression of stereo images: Preliminary results," *Signal Process.*, vol. 17, pp. 373–382, 1989.

[22] M. G. Perkins, "Data compression of stereopairs," *IEEE Trans. Commun.*, vol. 40, no. 4, pp. 684–696, Apr. 1992.

[23] H. Aydinoğlu and M. H. H. III, "Compression of multi-view images," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Austin, TX, Nov. 1994, pp. 385–389.

[24] H. Aydinoğlu, F. Kossentini, and M. H. H. III, "A new framework for multi-view image coding," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, vol. 4, Detroit, MI, May 1995, pp. 2173–2176.

[25] H.-Y. Shum and L.-W. He, "Rendering with concentric mosaics," in *Proc. SIGGRAPH*, Aug. 1999, pp. 299–306.

[26] H.-Y. Shum, K.-T. Ng, and S.-C. Chan, "Virtual reality using the concentric mosaic: Construction, rendering and data compression," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, Vancouver, Canada, Sep. 2000, pp. 644–647.

[27] C. Zhang and J. Li, "Compression and rendering of concentric mosaics with reference block codec (RBC)," in *Proc. SPIE Visual Communications and Image Processing* Perth, Australia, Jun. 2000, pp. 43–54.

[28] W.-H. Leung and T. Chen, "Compression with mosaic prediction for image-based rendering applications," in *Proc. IEEE Int. Conf. Multimedia Expo*, vol. 3, New York, Jul. 2000, pp. 1649–1652.

[29] K.-T. Ng, S.-C. Chan, and H.-Y. Shum, "Scalable coding and progressive transmission of concentric mosaic using nonlinear filter banks," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Thessaloniki, Greece, Oct. 2001, pp. 113–116.

[30] L. Luo, Y. Wu, J. Li, and Y.-Q. Zhang, "3-D wavelet compression and progressive inverse wavelet synthesis rendering of concentric mosaic," *IEEE Trans. Image Process.*, vol. 11, no. 7, pp. 802–816, Jul. 2002.

[31] X. Tong and R. M. Gray, "Coding of multi-view images for immersive viewing," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 4, Istanbul, Turkey, Jun. 2000, pp. 1879–1882.

[32] ——, "Interactive view synthesis from compressed light fields," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Thessaloniki, Greece, Oct. 2001, pp. 7–10.

[33] C. Zhang and J. Li, "Compression of lumigraph with multiple reference frame (MRF) prediction and just-in-time rendering," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 2000, pp. 253–262.

[34] P. Ramanathan, M. Flierl, and B. Girod, "Multi-hypothesis disparity-compensated light field compression," in *Proc. IEEE Int. Conf. Image Process.*, vol. 2, Thessaloniki, Greece, Oct. 2001, pp. 101–104.

[35] G. Miller, S. Rubin, and D. Ponceleon, "Lazy decompression of surface light fields for precomputed global illumination," *Proc. Eurographics Workship Rendering*, pp. 281–292, Aug. 1998.

[36] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, "Surface light fields for 3D photography," in *Proc. SIGGRAPH*, Aug. 2000, pp. 287–296.

[37] 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 *Proc. SIGGRAPH*, Jul. 2002, pp. 447–456.

[38] P. Eisert, E. Steinbach, and B. Girod, "Automatic reconstruction of stationary 3-D objects from multiple uncalibrated camera views," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 2, pp. 261–277, Mar. 2000.

[39] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM J. Math. Anal.*, vol. 29, no. 2, pp. 511–546, 1998.

[40] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 245–267, 1998.

[41] J. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang, "Memory-constrained 3-D wavelet transform for video coding without boundary effects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 9, pp. 812–818, Sep. 2002.

[42] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Commun. Pure Appl. Math.*, vol. 45, pp. 485–560, 1992.

[43] *ISO/IEC 15 444-1:2000*, 2002. JPEG 2000 Image Coding System.

[44] F. W. Wheeler and W. A. Pearlman, "Low-memory packetized SPIHT image compression," in *Proc. Conf. Record 33rd Asilomar Conf. Signals, Systems, Computers.*, vol. 2, Pacific Grove, CA, Oct. 1999, pp. 1193–1197.

[45] K. K. Lin and R. M. Gray, "Video residual coding using SPIHT and dependent optimization," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 2001, pp. 113–122.

[46] S. Cho and W. Pearlman, "3-D wavelet coding of video with arbitrary regions of support," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 3, pp. 157–171, Mar. 2002.

[47] B. Girod, C.-L. Chang, P. Ramanathan, and X. Zhu, "Light field compression using disparity-compensated lifting," presented at the IEEE Int. Conf. Acoustics, Speech, Signal Processing, Hong Kong, Apr. 2003.

[48] M. Held and R. M. Karp, The traveling salesman problem and minimum spanning trees: Part II, mathematical programming 1, 1971.

[49] S. Timsjo, "An Application of Lagrangian Relaxation to the Traveling Salesman Problem," Dept. Math. Phys., Malardalen Univ., Sweden, Tech. Rep. IMa-TOM-1999-02, Jun. 1999.

[50] S. Li and W. Li, "Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 8, pp. 725–743, Aug. 2000.

[51] G. Minami, Z. Xiong, A. Wang, and S. Mehrotra, "3-D wavelet coding of video with arbitrary regions of support," *Trans. Circuits Syst. Video Technol.*, vol. 11, no. 9, pp. 1063–1068, Sep. 2001.

[52] *ISO/IEC 11 544:1993*, 1993. Progressive Bi-level Image Compression.

[53] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.

[54] C.-L. Chang and B. Girod, "Receiver-based rate-distortion optimized interactive streaming for scalable bitstreams of light fields," presented at the Int. Conf. Multimedia Expo, Taipei, Taiwan, R.O.C., Jun. 2004.

[55] C.-L. Chang, X. Zhu, P. Ramanathan, and B. Girod, "Inter-view wavelet compression of light fields with disparity-compensated lifting," *Proc. SPIE Visual Communications Image Processing*, vol. 1, pp. 14–22, Jul. 2003.
[56] N. Mehrseresht and D. Taubman, "Adaptively weighted update steps in motion compensated lifting based scalable video compression," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Barcelona, Spain, Sep. 2003, pp. 771–774.
[57] B. Girod and S. Han, "Optimum motion-compensated lifting," *IEEE Signal Process. Lett.*, to be published.
[58] C.-L. Chang, X. Zhu, P. Ramanathan, and B. Girod, "Shape adaptation for light field compression," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, Barcelona, Spain, Sep. 2003, pp. 765–768.
[59] T. Sikora and B. Makai, "Shape-adaptive DCT for generic coding of video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 1, pp. 59–62, Feb. 1995.
[60] P. Kauff and K. Schuur, "A shape-adaptive DCT with block-based DC separation and Delta-DC correction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 3, pp. 237–242, Jun. 1998.

Chuo-Ling Chang received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1998, and the M.S. degree in electrical engineering from the Information Systems Laboratory, Stanford University, Stanford, CA, where he is currently pursuing the Ph.D. degree in electrical engineering.

His research interests include scalable coding and streaming of multimedia data.

Xiaoqing Zhu received the B.E. degree in electronics engineering from Tsinghua University, China, in 2001, the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 2002, where she is currently pursuing the Ph.D. degree.

For the summer of 2003, she was with IBM Almaden Research Center, working on intelligent document extraction and analysis. Her research interests include routing and resource allocation for video transport over ad-hoc wireless networks and wavelet compression of light fields.

Ms. Zhu was a recipient of the National Semiconductor Fellowship from 2001 to 2005.

Prashant Ramanathan received the B.A.Sc. degree in systems design engineering from the University of Waterloo, Waterloo, ON, Canada, in 1997, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1999 and 2005, respectively. His doctoral dissertation was on the compression and interactive streaming of image-based rendering data sets.

His research interests include image and video compression, multimedia streaming, computer graphics, and computer vision. He is currently Member of Technical Staff at NetEnrich, Inc.

Bernd Girod (F'98) received the M.S. degree from the Georgia Institute of Technology, Atlanta, and the Engineering Doctorate from the University of Hannover, Hannover, Germany.

He is a Professor of electrical engineering and (by courtesy) computer science in the Information Systems Laboratory, Stanford University, Stanford, CA. He was a the Chaired Professor of Telecommunications, Electrical Engineering Department, University of Erlangen-Nuremberg, Nuremberg, Germany, from 1993 to 1999. His research interests are in the areas of networked media systems and video signal compression. Prior visiting or regular faculty positions include the Massachusetts Institute of Technology, Cambridge; the Georgia Institute of Technology; and Stanford University. He has been involved with several startup ventures as Founder, Director, Investor, or Advisor, among them Vivo Software, 8x8 (Nasdaq: EGHT) and RealNetworks (Nasdaq: RNWK). Since 2004, he has served as the Chairman of the new Deutsche Telekom Laboratories, Berlin, Germany.