Rate-Distortion Optimized Interactive Light Field Streaming

Prashant Ramanathan, Mark Kalman, and Bernd Girod, *Fellow, IEEE*

***Abstract*—High-quality, photorealistic image-based rendering datasets are typically too large to download entirely before viewing, even when compressed. It is more suitable to instead stream the required image data to a remote user who can start interacting with the dataset immediately. This paper presents an interactive light field streaming system and proposes packet scheduling for transmitting the encoded image data in a rate-distortion optimized manner. An interactive light field streaming system must have low user latency. The system presented in this paper predicts the future user viewing trajectory to mitigate the effects of the low-latency constraints. Experimental results show that view prediction can improve performance, and that this improvement is limited by the prediction accuracy. The proposed packet scheduling algorithm considers network conditions and rate-distortion cost, knowledge of sent and received images, and the distortion for a set of images, to optimize the rendered image quality for the remote user. Rate-distortion optimized scheduling can be implemented either at the receiver or the sender. It is shown that this rate-distortion optimized packet scheduling can significantly improve performance over a heuristic scheduling approach. Experimental results also show that the encoding prediction dependency structure affects streaming performance both through the compression efficiency of the encoding and also through any decoding dependencies that may be introduced.**

***Index Terms*—Image-based rendering, image coding, image communication, interactive streaming, light field coding, light field streaming, light fields, rate-distortion optimized streaming.**

I. Introduction

FROM fly-arounds of automobiles to 360° panoramic views of cities to walk-throughs of houses, 3-D content is increasingly becoming commonplace on the Internet. Current content, however, offers limited mobility around the object or scene, and limited image resolution. To generate high-quality photo-realistic renderings of 3-D objects and scenes, computer graphics has traditionally turned to computationally expensive algorithms such as ray-tracing.

Recently, there has been increasing attention paid to image-based rendering (IBR) techniques that require only resampling captured images to render novel views. Such approaches are especially useful for interactive applications because of their low rendering complexity. Early examples include 360° panoramas and Apple's Quicktime VR [1]. A *light field* [2], [3] is an image-based rendering dataset that allows for photo-realistic rendering quality, as well as much greater freedom in navigating around the scene or object. To achieve this, light fields rely on a large number of captured images, but only implicit depth information. Earlier work refers to a similar idea called a ray-space representation [4], [5]. Concentric mosaics [6] are similar to light fields, but they restrict movement to along the horizontal plane, and do not exhibit vertical parallax.

Other IBR techniques use explicit depth information to perform rendering, such as 3-D warping [7], layered-depth images [8], [9], view-dependent texture mapping [10], [11], and surface light fields [12].

In the datasets used in this paper, explicit depth information is used for rendering. The image data are captured by a hemisphere of cameras surrounding an object. This dataset is referred to generically as a light field. In unoccluded free-space, this 4-D dataset is parameterized as a 2-D array of images. Light field rendering allows the user to zoom in or out of the scene. In this paper, the view trajectories that are used stay near the hemisphere of capturing cameras. Thus, in spirit, this is very similar to the view-dependent texture mapping [10], [11].

The large amount of image data can make transmitting light fields from a central server to a remote user a challenging problem. The file sizes of certain large datasets can be in the tens of Gigabytes [13]. Even over fast network connections, it could take hours to download the raw data for a large light field. This motivates the need for efficient compression algorithms to reduce the file size.

Numerous algorithms have been proposed to compress light fields. These light field compression algorithms work by exploiting two types of correlation in the image dataset. The correlation between pixels in an image is usually captured using existing image coding techniques. The correlation between two or more nearby images in the dataset is exploited using a technique called disparity compensation, which is analogous to motion compensation in video, first used in stereo image coding [14]–[17].

Disparity compensation uses either implicit geometry information or an explicit geometry model, if available, to warp one image to another image. This allows for prediction between images, or encoding several images jointly by warping them to a common reference frame. This approach is taken in early work in multi-view coding [18], [19]. The class of closed-loop predictive (DPCM) light field coders includes [20]–[25]. The hierarchical disparity-compensating coder in [23], [24], which belongs to this class of algorithms, is used in this paper. Here, the

Manuscript received August 29, 2005; revised August 1, 2006. This work was supported in part by the National Science Foundation under Grant ECS-0225315. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jin Li.

P. Ramanathan is with AOL, LLC, San Francisco, CA 94140 USA (e-mail: pramanat@gmail.com).

M. Kalman and B. Girod are with Stanford University, Stanford, CA 94305 USA (e-mail: mkalman@stanford.edu; bgirod@stanford.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2007.893350

images are grouped into different levels of the prediction hierarchy. Images at the lowest level at encoded independently. Images at higher levels are encoded using disparity-compensated prediction from the lower levels.

Other approaches to light field coding include an open-loop, wavelet-transform approach, such as in [24], [26], [27], or more recently, using disparity-compensated lifting [25], [28], [29]. There are also some vector quantization approaches [2], [30]. This paper, however, only considers the closed-loop predictive approach mentioned earlier.

Transmitting light field data to a remote user has typically taken place in a download scenario, where the entire dataset is transmitted to the user before the user can start interacting with the dataset. This requires the user to wait for the download, and may even be infeasible for very large datasets or over limited bandwidth connections.

A more interesting and useful method of transmitting the dataset is to stream the required data, as it is needed by the user. In this scenario, the user can start interacting with the dataset almost immediately after starting the streaming session. For streaming, the encoded light field must be packetized, and appropriate packets must be transmitted. For error-prone channels such as the Internet, additional mechanisms to deal with lost or delayed packets must be considered.

For interactive streaming, the image data is transmitted with consideration of what the user wants to see at a particular instant. Region-of-interest (ROI) image coding and transmission, such as in the JPEG2000 standard [31]–[33], is related in spirit to interactive streaming.

Scalable light field coding algorithms [24], [26], [34], based on wavelet coding, have been proposed for progressive transmission of light field data. The work in [35], [36] focuses on the packetization and the bitstream assembly of similar scalable light field encodings. The work in [37] describes a streaming system for concentric mosaics encoded with a two-level prediction-based scheme [38]. This system assigns priority to image data in the lowest level of the hierarchy that is required to predict the data in the higher hierarchy levels, and does not consider additional rate-distortion criteria. In [39], [40], a general framework for representing many different classes of scalable multimedia data is proposed. One example application is the streaming of IBR datasets. A simple utility-cost measure is used to trade off between the various dimensions of scalability. The transmission approach used can leverage multicast and broadcast transmission mechanisms to reduce server bandwidth requirements when streaming to large numbers of clients.

This paper proposes a light field streaming framework that schedules images for transmission based on a rate-distortion criterion, improving on previous heuristic scheduling rules. This framework selects images for transmission based on the desired user viewing trajectory, feedback in the form of acknowledgements from the receiver, rate constraints, knowledge of sent and received images, and the rendered view quality for a set of images for the desired view trajectory. This work is based on the rate-distortion optimized streaming framework for audio and video data, summarized in Section II. The interactive light field streaming system is presented in Section III, focusing on the low-latency requirements of such a system. Rate-distortion optimized packet scheduling for the interactive light field streaming system is presented in Section IV. Experimental results evaluating the system are presented in Section V.

II. Background: R-D Optimized Streaming

Streaming over a best-effort network such as the Internet requires error control techniques and possibly error concealment at the client. Two common error control schemes are forward error correction (FEC), such as priority encoding transmission (PET) [41], and automatic retransmission request (ARQ) [42], [43]. A more sophisticated error control scheme involves scheduling more important packets earlier so that they are more likely to arrive. In [44], [45], a rate-distortion optimized framework for the streaming of media over a lossy packetized network is presented. It is summarized in this section. The framework assumes a compressed media representation that has been assembled into packets or data units. Associated with each data unit is the data unit size, for instance, in bytes, and the deadline by which the data unit must arrive in order for it to be useful for playout.

Because images used to render views are quantized and because some of the packets that transport the encoded images may not arrive in time, rendered views are distorted. In order to estimate the distortion in a computationally efficient manner, the authors in [44], [45] assume that each data unit contributes to reducing the distortion that the user experiences. Associated with each data unit is a distortion reduction value. This value is the amount that the distortion of the view will be reduced if the data unit is decodable when it is needed. A data unit is decodable if it arrives in time and all of the ancestor data units upon which it is dependent arrive in time as well. Distortion reductions are assumed to be additive, and the overall distortion is computed by using an acyclic directed graph that describes the inter-dependencies between the data units in the media presentation.

A transmission policy π_l is associated with each data unit l that, for instance, describes if and when a packet should be transmitted or retransmitted. For each given transmission policy π_l, there is an associated cost, $\rho(\pi_l)$, for each byte of the data unit and error probability $\epsilon(\pi_l)$, which is the probability that the data unit does not arrive by the playout deadline.

The goal is to determine the optimal transmission policies for all data units, $\boldsymbol{\pi} = [\pi_1\ \pi_2\ \cdots\ \pi_N]$, given per-byte costs, deadlines and distortion reductions, along with the knowledge of network packet loss and delay characteristics, acknowledgements from the receiver, and transmission history. The policy that minimizes the overall rate-distortion Lagrangian cost

$$J(\boldsymbol{\pi}) = D(\boldsymbol{\pi}) + \lambda R(\boldsymbol{\pi}) \tag{1}$$

is selected as the optimal policy. The parameter λ controls the tradeoff between rate and distortion.

The expected rate $R(\boldsymbol{\pi})$ depends upon the data unit sizes B_l and the expected number of transmissions $\rho(\pi_l)$. The expected distortion $D(\boldsymbol{\pi})$ depends upon the error probabilities $\epsilon(\pi_l)$, the distortion reduction values and the inter-dependency graph for the data units.

With a large number of data units, it is not tractable to exactly solve the minimization problem in (1). A reasonable approximate solution is to find the optimal policy for each data unit,

while keeping the policies for all other data units fixed, and iterate until the overall solution converges. While the solution is guaranteed to converge, it may only converge to a local minima [44], [45]. Writing the cost function in (1) with all policy variables held fixed except that of data unit l yields the following cost function which is in terms of only variable pi_l

$$J_l(\pi_l) = \epsilon(\pi_l) + \lambda' \rho(\pi_l) \tag{2}$$

where $\epsilon(\pi_l)$ is the error probability and $\rho(\pi_l)$ is the per-byte cost as before. It can be shown that $\lambda' = \lambda B_l / S_l$. This incorporates the rate-distortion tradeoff operator λ from (1), the data unit size B_l, and S_l, the sensitivity of the overall distortion to not having received data unit l by its deadline. The sensitivity term represents the relative importance of a particular data unit.

The expected number of transmissions $\rho(\pi_l)$ and the expected error probability $\epsilon(\pi_l)$ can be calculated using the specified loss probability and delay probability density function for the forward and back channels. The expected cost, for instance, must take into account that the receipt of an acknowledgement would abort any further transmissions and, in general, reduce the cost. The expectation for the error probability must be taken over this space of truncated transmission policies.

III. System Overview

A light field streaming system transmits images from the *sender*, where the image data initially resides, to the *receiver*, where the remote user views and interacts with the light field. In a light field streaming session, the receiver initiates the session, and the sender responds by sending preliminary data about the light field, such as a geometry model for rendering, light field camera view parameters and the focal length, dimensions of the images, and other intrinsic camera parameters. In a receiver-driven scenario, the sender additionally must send rate-distortion preamble data, described in more detail in Section IV-B.

The user can then begin selecting the desired view point, with a mouse or other pointing device. This desired view must be rendered on the display device within an acceptable amount of time. Typically, tolerable latencies are in the range of 100 ms to 200 ms. The maximum one-way delay for telephony or video conferencing recommended by the ITU is 150 ms [46]. For interactive computer graphics applications, the frame rate or rendering rate is another important metric. For interactivity, frame rates need to be at least 20 Hz. This corresponds to rendering every 50 ms. This value is used throughout all streaming experiments in this paper.

For a remote viewing scenario, it is often not possible to send a request and transmit images within the deadlines imposed by the latency constraints. This is mitigated by predicting the user's future viewing trajectory, described in Section III-B. The entire system, with view prediction, is described next.

A. Transmission Protocol

Fig. 1 shows a diagram of the set of actions taken by the sender and the receiver for receiver-driven packet scheduling. The sender-driven system is similar, but is not described due to space constraints.

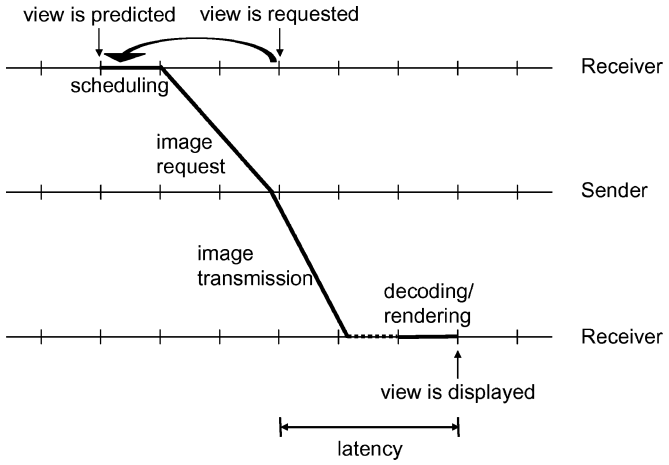

Fig. 1. Receiver-driven transmission protocol. The receiver predicts the view, schedules the images for requests, and transmits image requests to the sender. The sender immediately transmits images to the receiver. The receiver decodes and renders the view, displaying it within the deadline given by the user latency constraint.

The first step of the streaming process is predicting the view trajectory. Essentially, it can be thought of as a way of knowing a desired view, approximately, ahead of time. To make this idea more precise, suppose that the user wants to view the trajectory $\mathbf{v} = \{v_1, v_2, \cdots, v_M\}$, where M is the number of views in the trajectory. Each view v_i is requested at time t_{v_i}, relative to a global clock, with regular intervals $t_{v_{i+1}} - t_{v_i} = \Delta t_{\text{view}}$. It is assumed that the sender and receiver clocks are synchronized, and therefore, all times can be given in terms of a single global reference clock. If the view trajectory can be predicted ahead by Δt_{pred}, then view $\tilde{v}_i$, a prediction of desired view v_i, can be known at time $t_{v_i} - \Delta t_{\text{pred}}$ instead of time t_{v_i}.

This prediction of the view trajectory is implemented at the receiver. In Fig. 1, view prediction is denoted by the backwards-pointing arrow on the top "receiver" line of the diagrams. Here, having predicted the view trajectory, the desired images can be requested.

Scheduling generates a set of image requests. Images requests are sent from the receiver at regular intervals called request opportunities. To simplify the explanation of the system, the interval between request opportunities Δt_{oppty} is the same as the interval between desired view points Δt_{view}. Scheduling is assumed to take a fixed amount of time Δt_{sched}, indicated on the top "receiver" line on Fig. 1.

The image requests are transmitted from receiver to the sender, taking an interval of time Δt_{req}, and the sender responds immediately by transmitting an image packet, which takes time interval Δt_{xmit} to arrive at the receiver. The time from image request to receiving the data $\Delta t_{\text{req}} + \Delta t_{\text{xmit}}$ is a random quantity that depends on the network. Once the client receives the image packets, it decodes them and renders the view, taking times Δt_{dec} and Δt_{render}, respectively. Note that responding to requests by transmitting the appropriate image data is the only task that the sender must perform. Therefore, it can easily serve many clients simultaneously.

The image data for a particular view in the view trajectory must arrive within a fixed period of time after the user selects

it. Suppose that the maximum tolerable latency between having selected a view and it being rendered is $L_{\max}$. Therefore, view v_i must be completely rendered on the display screen by $t_{v_i} + L_{\max}$.

This latency constraint, along with the lookahead of prediction, determines which views in the trajectory are to be considered for scheduling. A view v_i is considered for scheduling at time t_s if it can potentially arrive in time to be rendered. In other words, the scheduler considers views for all indexes i such that $t_s + \Delta t_{\text{sched}} + \Delta t_{\text{req}} + \Delta t_{\text{xmit}} + \Delta t_{\text{dec}} + \Delta t_{\text{render}} < t_{v_i} + L_{\max}$. The processing times for scheduling, decoding and rendering can be combined into one term $\Delta t_{\text{proc}} = \Delta t_{\text{sched}} + \Delta t_{\text{dec}} + \Delta t_{\text{render}}$.

A view v_i can only be considered if it is known, or known at least approximately through prediction. At scheduling time t_s, views for all indexes i for which $t_{v_i} - \Delta t_{\text{pred}} < t_s$ are known or can be estimated. Combining these two conditions gives the boundaries of the view window. At scheduling time t_s, the view window consists of all views with indexes i such that

$$t_s + \Delta t_{\text{proc}} - L_{\max} < t_{v_i} \leq t_s + \Delta t_{\text{pred}} \tag{3}$$

using the fact that image request and transmission times are random variables which, in general, only have the restriction that they are positive: $\Delta t_{\text{req}} + \Delta t_{\text{xmit}} > 0$.

B. View Prediction

Accurate prediction of the user's future view trajectory is an essential component of the streaming system. View trajectory prediction has been used to provide a low latency experience in Virtual Reality environments, as in [47]. It has also been extensively used in networked multi-player video games, investigated, for instance, in [48]. The simplest, and most widely used technique for these scenarios, dead reckoning, extrapolates future views by assuming that the users maintain their current velocity.

The rendering system that is used to generate the experimental results in this paper allows for the user to select the view position using a computer mouse as the input device [49]. The prediction of the view trajectory can exploit knowledge and access to this rendering system by doing the prediction in the 2-D mouse move space instead of 3-D view coordinates. Since the user input originates in 2-D mouse move space, it follows that prediction should be performed in this space, taking into account the characteristics and constraints of such user input. Also, it is simpler to predict a trajectory in 2-D space instead of 3-D space.

When navigating with the system in [49], mouse moves tend to have many straight lines, so they are amenable to using a simple prediction technique. In this work, dead reckoning is used with an autoregressive moving average (ARMA) filter to smooth out temporal fluctuations in the velocity estimate. Improving the view prediction scheme, especially by considering the nature of user interaction, is an interesting avenue for future investigation.

The mouse position at time index k is denoted by $p^{(k)} = (x^{(k)}, y^{(k)})$, for $k = 0, 1, \cdots$. The estimated velocity

$$\tilde{v}^{(k)} = \alpha \left(p^{(k)} - p^{(k-1)} \right) + (1 - \alpha) \tilde{v}^{(k-1)} \tag{4}$$

at time k combines both the current and previous velocity estimates. Decreasing the parameter α increases the temporal smoothness of the velocity estimate. A value of $\alpha = 0.5$ was empirically found to give the best performance in terms of predicting future mouse positions.

If the mouse positions are known up to time index k, then the velocity estimate $\tilde{v}^{(k)}$ can be computed recursively. In practice, the velocity is simply updated every time a new mouse position becomes available. The future mouse position for time index $m > k$ is computed using

$$\hat{p}^{(m)} = p^{(k)} + (m - k) \tilde{v}^{(k)}. \tag{5}$$

An implicit assumption in (4) and (5) is that the time interval between consecutive time indexes is constant.

Any number of time steps can be computed with (4) and (5), but it is typically not possible to accurately look very far into the future, depending on the nature of the user motion. Given predicted future mouse moves, using the same mapping that the rendering program uses to map each mouse move to a new viewing position, the future view trajectory can be predicted.

IV. Rate-Distortion Optimized Light Field Streaming

In order to adapt the framework described in Section II [44], [45] to light field streaming, the packet scheduling algorithm must explicitly take into account the interactivity of the application, and send the image data based on the current desired user view trajectory. Three concepts are incorporated to adapt the framework of Chou *et al.* to light field streaming: view-dependent distortion, multiple deadlines, and state-based distortion. These ideas are used to derive the distortion and sensitivity calculation for packet scheduling, and have been described in [50].

The rate-distortion optimized streaming framework attempts to minimize the Lagrangian cost $D(\boldsymbol{\pi}) + \lambda R(\boldsymbol{\pi})$ (1), as described in [44] and [45]. The policy $\boldsymbol{\pi} = (\pi_1, \pi_2, \cdots, \pi_L)$, where π_l refers to the policy for a data unit l, is used for both the receiver-driven and sender-driven scenarios. In a sender-driven scenario, the policy π_l indicates whether or not the data unit l is sent at each of the transmission opportunities considered in the scheduling window. In the receiver-driven scenario, the policy π_l indicates whether the data unit l is requested at each of the request opportunities considered in the scheduling window.

A. State-Dependent Distortion

One of the important contributions of Chou *et al.* [44], [45] is to simplify the distortion calculation by assuming additive distortion and introducing a dependency graph to explicitly account for the decoding dependencies. This model, however, cannot be used for light field images that may have no decoding dependency but might be highly correlated with one another. For instance, an image may be very important in terms of distortion if none of its neighboring images are available at the receiver, but only slightly important if one neighboring image arrives.

By considering all combinations of images, a more accurate estimate of the distortion can be derived. Suppose that for a particular view $v(t_i)$ at time instance t_i, a set of L data units, $\mathcal{L}_{v(t_i)}$, is required to render that view. There are 2^L possible combinations of arrivals and non-arrivals for this set of data units. Let each combination of available and unavailable packets be

called a packet availability state s, that belongs to the state space $\mathcal{P}_{v(t_i)}$. For every $s \in \mathcal{P}_{v(t_i)}$, the corresponding distortion that an end-user would experience can be computed, resulting in 2^L distortion values. These distortion values are used to compute the expected distortion for a transmission policy.

This state-based approach can quickly become intractable for even a modest number of data units. For views along the camera plane, only a few images may be required for rendering, but many more images are needed due to the prediction structure. This very same hierarchical prediction structure, however, can be exploited to reduce the state space.

According to the hierarchy, the data units can be organized into several different levels, where the data units at a particular level depend upon some of the data units at next lower level. To simplify the state-space, it is assumed that a data unit requires all the data units at the next lower level, which is true for many of the trajectories and prediction dependency structures used.

All combinations of arrivals and non-arrivals of data units at a particular level are considered only if all data units at all lower levels have arrived. The resulting total number of states is less than the sum of the number of states for the data units for each level. If the data units are evenly distributed across levels, then this results in a significant reduction in the size of the state space. This reduced state space is denoted as $\mathcal{P}'_{v(t_i)}$. A distortion value is computed for each state $s \in \mathcal{P}'_{v(t_i)}$.

B. View-Trajectory-Dependent Distortion

Rate-distortion optimized packet scheduling depends upon accurately calculating the rate for sending a particular set of image packets, and accurately estimating the distortion when using a set of image packets to render a particular view of the light field. Unlike in video or audio, the distortion depends upon the user's viewing trajectory. Hence, in the light field streaming system, distortion is parameterized by the view v.

In theory, the distortions for every combination of possible views and set of image packets need to be computed and stored. In practice, however, this is intractable and these values are interpolated. For receiver-driven streaming, this distortion information also needs to be transmitted to the receiver initially and, thus, must be compactly represented.

To estimate distortion, the idea of organizing data units into levels, discussed in Section IV-A, is exploited again by estimating the distortion in two stages. First, a distortion $D_{v,l}$ is defined for a view v and level l. This quantity is defined as the rendered distortion where all the images that belong to level l and below are available, but those for levels higher than l are not. If there are L levels in a light field, then $D_{v,l}$ for $l = 0, \cdots, L$ can be defined, where $D_{v,0}$ is the distortion when no images are available.

In general, two neighboring views v_1 and v_2 may require different sets of images for rendering. The quantity $D_{v,l}$, however, is defined for all views v on the hemisphere and for all levels $l = 0, \cdots, L$. Also, experiments not reported in this paper show, for fixed level l, $D_{v,l}$ varies smoothly over the hemisphere of views v. This property makes it amenable to view interpolation.

$D_{v,l}$ is calculated for a fixed set of views $v \in \mathcal{V}$, and $l = 0, \cdots, L$. In the experiments in this paper, 1000 views on the hemisphere of views are used. To calculate $D_{v,l}$ for a view $v \notin \mathcal{V}$, a weighted average of the distortion values from nearby available distortion values is used, using angular distance between viewpoints to determine the weights.

The interpolated values $D_{v,l}$ can now be used to compute the distortions for the states belonging to level l. These states correspond to the case where all images for levels higher than l are not available, while all images for levels lower than l are available. This set of states is denoted as $\mathcal{P}'_{v,l}$. The variable $s^{l,0} \in \mathcal{P}'_{v,l}$ represents the state where no images are available in level l and the variable $s^{l,1} \in \mathcal{P}'_{v,l}$ represents the state where all images belonging to level l are available. It should be noted that $D(s^{l,0}, v) = D_{v,l-1}$ and $D(s^{l,1}, v) = D_{v,l}$. The distortion $D(s, v)$ for $s \in \mathcal{P}'_{v,l}$ must be derived.

One approach to computing the distortion is using the theoretical framework described in [51], [52]. The theoretical framework considers the statistical properties of the images, correlation between images and the encoding scheme, to estimate the distortion, denoted by $D^t(s, v)$. Due to the numerous simplifying assumptions made in the theoretical model, the theoretical distortion values $D^t(s, v)$ are not accurate enough to use directly. They can, however, be used to derive a scaling function and used in conjunction with the computed values $D_{v,l-1}$ and $D_{v,l}$.

The scaling function is

$$\alpha_{v,l}(s) = \frac{D^t(s, v) - D^t(s^{l,0}, v)}{D^t(s^{l,1}, v) - D^t(s^{l,0}, v)} \tag{6}$$

where $D^t(s, v)$ represents the distortion derived from theory. This scaling function can be used to relate the distortion $D(s, v)$ to the known quantities $D(s^{l,0}, v)$ and $D(s^{l,1}, v)$. The estimated distortion is given by

$$\hat{D}(s, v) = \alpha_{v,l}(s)(D_{v,l} - D_{v,l-1}) + D_{v,l-1}. \tag{7}$$

In order to use these estimated distortions in a receiver-driven scenario, the view-dependent distortion values need to be efficiently represented and transmitted. The distortion values $D_{v,l}$ are computed only once for each light field. Each of these values is quantized with a fixed length code and the minimum and maximum of these values is also stored. The quantized values represents the position in the range between the minimum and maximum values.

C. Multiple Deadlines

In the framework of Chou *et al.*, a data unit is associated with one particular instance in time, its playout deadline. Light fields, on the other hand, typically require a particular data unit to render several different views along the user view trajectory, at different time instances. Hence, there are multiple rendering deadlines that are associated with a data unit. In order to incorporate this into the streaming framework, there are several changes that must be made, starting with the distortion calculation.

The distortion that the user experiences is defined in this paper as the sum of the distortions for each time instance, or view, in the viewing trajectory, over some viewing window. The complete expression for distortion is given below in (10). The viewing window is the set of views that are considered when

performing the packet scheduling, and can be thought of as the view trajectory that can be predicted by the view trajectory prediction module.

By defining the distortion over a viewing window, a data unit may be associated with multiple playout deadlines. This means that a data unit will have multiple error probabilities, $\{\epsilon(\pi_l, t_i)\}$, that is, the probability of data unit l not arriving by time t_i given transmission policy π_l, corresponding to each of the playout deadlines t_i. The expected number of transmissions for a data unit $\rho(\pi_l)$, however, is still the same. With these changes, the minimization procedure also needs to be modified. The policy for each data unit is still minimized independently in an iterative fashion as before, but with a slightly different cost function. The cost function for each data unit is modified from (2) to become

$$J_l(\pi_l) = \rho(\pi_l) + \sum_{i \in \mathcal{T}} \nu_{t_i} \epsilon(\pi_l, t_i) \tag{8}$$

where all the time instances t_i in the viewing window, indexed by $i \in \mathcal{T}$ are considered. The quantity ν_{t_i}, given by $\nu_{t_i} = S_{l,t_i}/\lambda B_l$, is analogous to the reciprocal of λ' in (2). Note that the sensitivity term S_{l,t_i} is also indexed by time; it is the sensitivity of the overall distortion to the non-arrival of data unit l by time instance t_i.

The cost (8) is computed for each policy, and the one with the lowest cost is selected as the optimal policy. Since it is the policy length that tends to determine the complexity of the algorithm, using multiple deadlines does not significantly impact the computational complexity of the algorithm.

D. Distortion and Sensitivity Calculation

The objective in rate-distortion optimized packet scheduling is to find the policy $\boldsymbol{\pi}$ to minimize the Lagrangian rate-distortion cost given by

$$J(\boldsymbol{\pi}; \mathbf{v}) = D(\boldsymbol{\pi}; \mathbf{v}) + \lambda R(\boldsymbol{\pi}) \tag{9}$$

for view trajectory $\mathbf{v}$ and Lagrangian parameter λ.

Combining the concepts of view-trajectory-dependent distortion, multiple deadlines, and state-dependent distortion described in previous sections, the expected distortion is given by

$$D(\boldsymbol{\pi}; \mathbf{v}) = \sum_{i \in \mathcal{T}} \left[\sum_{s \in \mathcal{P}'_{v(t_i)}} D\left(s, v(t_i)\right) \Pr\{s\} \right]. \tag{10}$$

If transmissions of data units are considered to be independent over the channel, then the probability $\Pr\{s\}$ of state s is given by

$$\Pr\{s\} = \prod_{\substack{l \in \mathcal{L}_{v(t_i)} \\ l: s_l = 1}} (1 - \epsilon(\pi_l, t_i)) \prod_{\substack{l \in \mathcal{L}_{v(t_i)} \\ l: s_l = 0}} (\epsilon(\pi_l, t_i)) \tag{11}$$

where the binary-valued variable s_l indicates whether data unit l has arrived or not in state s. The state s also includes data units that are known to have been successfully received. Thus, the estimated distortion in (10) accounts for data units that have been received. The rate, given by

$$R(\boldsymbol{\pi}) = \sum_l B_l \rho(\pi_l) \tag{12}$$

does not depend on the view trajectory directly. Note, however, that it is coupled indirectly through the choice of the transmission policy $\boldsymbol{\pi}$ appropriate for that trajectory.

Minimizing the rate-distortion cost (9) is performed iteratively, by considering only the policy of one data unit at a time. The minimization then simplifies to minimizing the expression in (8). The Lagrangian parameter $\nu_{t_i} = S_{l,t_i}/\lambda B_l$ trades off the probability of the data unit not arriving by deadline t_i, $\epsilon(\pi_l, t_i)$, against the expected number of transmissions, $\rho(\pi_l)$. Note that this term includes the original Lagrangian tradeoff parameter λ as well as the data unit size B_l. The sensitivity is derived from (10) and (11) to give

$$S_{l,t_i} = \begin{cases} \sum_{\substack{s \in \mathcal{P}'_{v(t_i)} \\ s_l = 0}} S'_{s,l,t_i} - \sum_{\substack{s \in \mathcal{P}'_{v(t_i)} \\ s_l = 1}} S'_{s,l,t_i} & \text{if } l \in \mathcal{L}_{v(t_i)}, \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

where

$$S'_{s,l,t_i} = D\left(s, v(t_i)\right) \prod_{\substack{l' \in \mathcal{L}_{v(t_i)} \\ l': s_{l'} = 1 \\ l' \neq l}} (1 - \epsilon(\pi_{l'}, t_i)) \prod_{\substack{l' \in \mathcal{L}_{v(t_i)} \\ l': s_{l'} = 0 \\ l' \neq l}} (\epsilon(\pi_{l'}, t_i)). \tag{14}$$

The distortion $D(s, v(t_i))$ for state s and view $v(t_i)$ is calculated using the techniques described in Sections IV-A and IV-B. The error probabilities $\epsilon(\pi_{l'}, t_i)$ that appear in the minimization expression (8) and the sensitivity term (14), along with the cost $\rho(\pi_l)$ that appears in (8), are calculated in the same way as in Chou *et al.* [44], [45]. That is, the expected error probability or expected cost is based on the loss probability and delay distribution of the transmission channel.

For the error probabilities, an expression is computed for each deadline t_i, but this calculation is identical to Chou *et al.*, except that the length of the policy that is considered may be shortened according to the deadline. The calculation of the expected cost does not change. These calculations are slightly different for the sender-driven and receiver-driven scenarios, since sender-driven streaming consider transmission policies, and receiver-driven streaming considers request policies [44], [45].

E. Heuristic Packet Scheduling

Packet scheduling can also be done in a heuristic manner. This section presents an ARQ scheme for light field streaming. The task is to decide which images to transmit at each transmission opportunity, for a sender-driven scenario, or which image to request at each request opportunity, for a receiver-driven scenario. The method described below is equally applicable to both the sender-driven and receiver-driven scenarios.

It is assumed that the scheduler knows the maximum rate for sending data units, either through feedback from the network or receiver, or because it has been explicitly specified. The task then is to select which images to send at each transmission opportunity, while not violating the bit-rate constraint. One way to do this is, for each transmission opportunity, to assign a priority to each of the data units in the current viewing window. Data

units are transmitted in order of priority until the bit budget for that transmission opportunity is exhausted.

In the heuristic scheme, the priority of a data unit for transmission is determined by four factors. The first is whether the data unit has already been sent. Data units that have not yet been sent have priority over those that have already been sent. A data unit may also get lost. In order to account for this, when the probability that the data unit is lost, given that it has not yet been acknowledged, reaches a certain threshold, it is considered to be lost. A threshold of 99% is used in the experiments. Lost packets are given the same priority as packets that have not been sent.

The second criterion dictates that data units that are lower in the hierarchy and are used for prediction are given priority over those higher in the hierarchy. The third criterion assigns priority according to whether the data unit is needed sooner for rendering. The data unit with the earlier playout deadline is given higher priority.

The final criterion for assigning priority is based on the viewpoint of the data unit. The data unit whose viewpoint is closest to the viewpoint associated with the view of the earliest playout deadline is given priority. Since a hemispherical viewing arrangement is used, the closeness of the viewpoint can be determined by measuring angular distance between the two views.

These criteria are used in succession. For instance, the third criterion is only used to decide between data units with the first two criteria identical.

V. Experimental Results

This section describes experiments that examine the performance of the light field streaming system. In Section V-A, rate-distortion optimized packet scheduling is compared to heuristic packet scheduling. In Section V-B, the effects of predicting the user's future view trajectory are examined. In Section V-C, the effect of the encoding prediction dependency structure on streaming performance is investigated.

Streaming experiments have been performed on several datasets, but due to space considerations, are only shown for two representative datasets: *Bust* and *Star*. Full experimental results are available in [53]. The *Bust* light field contains 339 images, each of resolution 768 $\times$ 480, and the *Star* light field contains 281 images, each of resolution 768 $\times$ 512. These datasets are encoded with 3 different encodings, INTRA, PRED2 and PRED4, using closed-loop disparity-compensated prediction. INTRA encoding refers to independent encoding of the images. PRED2 and PRED4 use 2 and 4 image prediction levels, respectively. In PRED4, for instance, the images are grouped into 4 levels. Images in the first level are independently encoded, images in the second level are predicted from those in the first level, images in the third level are predicted from those in the first two levels, and so on. A similar scheme is for PRED2. In all encodings, the images are encoded with a quantization parameter of $Q = 3$, which represents high image quality.

The streaming experiments are performed by simulating an i.i.d. forward and back channel. For each data unit, acknowledgement, request, or view trajectory that is to be sent, the loss and delay of this transmission is drawn as follows. A loss is drawn from a Bernoulli distribution, with probability 0.1%. The delay distribution is assumed to be a shifted gamma distribution as in [44], [45]. Using the same notation, the delay distribution for a data unit, given that it is not lost is

$$f_T(\tau) = \begin{cases} \frac{\alpha(\alpha(\tau-\kappa))^{n-1}e^{-\alpha(\tau-\kappa)}}{\Gamma(n)} & \text{if } \tau \geq \kappa; \\ 0 & \text{otherwise.} \end{cases} \quad (15)$$

The quantity κ represents a fixed, deterministic delay, and the parameters n and α determine the mean and variance of the distribution. In the following experiments, $n = 2$, $\kappa = 25$ ms and $\alpha = 0.04$ $(1/\text{ms})$, which corresponds to a mean delay of 50 ms and a standard deviation of 17.7 ms. The parameters that are used represent a round-trip delay of 100 ms, where 50 ms of that are due to the fixed delay κ. This channel could represent, for instance, a cross-continental link.

For rate-distortion optimized streaming, different points on the rate-distortion curve are computed by sweeping the Lagrangian parameter λ from 0 to ∞. For heuristic streaming, the average target bit-rate is set for different desired rate points. For each set of experiment parameters, 10 independent network simulations are used and averaged to give the final result. The scheduling algorithm decides which images to transmit, according to the available information.

The rate and distortion are calculated for each simulation over the network, and for each rate-distortion tradeoff point. The rate is calculated by adding the sizes of all the transmitted data units and dividing by the total transmission time and is reported in kbps. The rendered distortion is computed in terms of luminance MSE distortion by comparing a rendered view using the available reconstructed images, to the rendered view using the original uncompressed light field. This MSE distortion is converted to PSNR and is averaged in terms of PSNR over the entire desired user view trajectory. The average PSNR value is then reported.

The trajectories that are used in the experiments have been captured with the light field viewer in [49]. The viewing positions of these trajectories are typically near the hemisphere of the viewing positions of the capturing light field cameras. This allows for a view to be rendered with a relatively small number of images. Typically, 4 images are used. Such an approach is very similar to the view-dependent texture mapping approach [10], [11]. Trajectories away from the hemisphere of capturing camera positions, either zooming in or out, may require many images to render a single view. Such trajectories have not been explored in this paper.

For each dataset, ten trajectories each, from three types of trajectories are captured. The first type of trajectories, called *slow*, contain slow, deliberate movements by the user. The second type of trajectories, called *medium*, contain faster, more typical movements by the user. The third type of trajectories, called *fast* contain rapid, often erratic, movements by the user. All of the trajectories consist of rotational motion around the object.

Fig. 2 shows an example of each type of trajectory, relative to the camera positions and the object, for another example light field. The camera views and directions are indicted by x's and the viewing trajectory is illustrated with circles indicating view positions. The geometry of the object is shown in the center

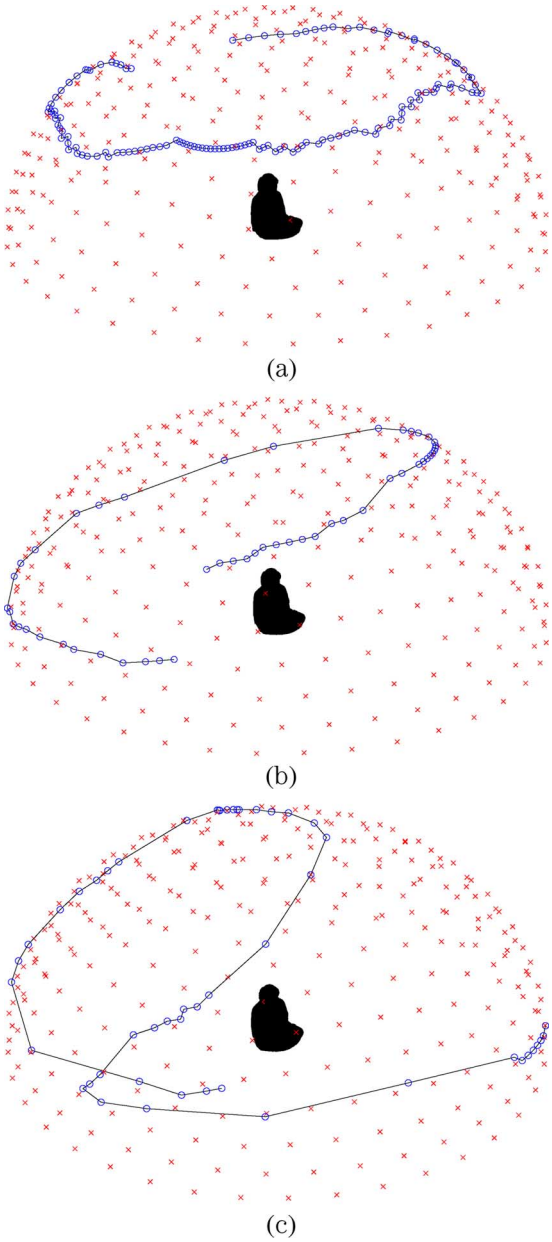

Fig. 2. Illustrations of the (a) *slow*, (b) *medium*, and (c) *fast* trajectories for an example light field.

of the hemisphere. Each view trajectory consists of 50 views, occurring every 50 ms, for a total trajectory time of 2.5 s.

In the following sections, the rate-distortion performance is averaged over the 10 trajectories of each type of trajectory. These results are characterized by the type of user view trajectory.

A. RD-Optimized versus Heuristic

The streaming performance using rate-distortion optimized packet scheduling is compared to that of heuristic scheduling for the *Bust* and *Star* datasets, all encodings (INTRA, PRED2, PRED4), and for the *medium* trajectory. A prediction lookahead of 200 ms is used with a processing time of 100 ms.

Figs. 3 and 4 show the results for the *Bust* and *Star* light fields, respectively. At low rates, very few or no data units are sent, giving similar performance for both the rate-distortion optimized and heuristic scheduling systems. At high rates, all the data units in the viewing window are sent, again resulting in similar performance for both the rate-distortion optimized and heuristic streaming approaches. The interesting region is in the remaining rate region, which is shown in these figures.

Rate-distortion optimized scheduling consistently outperforms heuristic scheduling for all light fields and encodings. Results for the other light fields, not presented here due to space considerations, show similar trends [53]. In the middle rate region, the performance improvement is greatest. An increase in PSNR of up to 5 dB or more, at the same bit-rate, is observed.

Rate-distortion optimized packet scheduling outperforms the heuristic for a few reasons. Rate-distortion optimized scheduling considers the probability of a data unit arriving by its decoding deadline, and avoids transmitting images that are likely to be late. Also, by considering the distortion and rates for a set of images, for all the views in the current view window, rate-distortion optimized packet scheduling can make more intelligent choices about which images to send. It often sends different images than the heuristic scheme.

Finally, the heuristic scheme uses "leaky-bucket" rate-control. Rate-distortion optimized streaming, however, controls only the average rate, giving it more flexibility. As a result, its transmissions tend to be more bursty. The burstiness is limited to only a few images in a transmission opportunity, because the scheduling view window is typically small. Precise rate control with rate-distortion optimized streaming, however, can be difficult, as discussed in [44], [45].

B. View Prediction and Prediction Lookahead

Prediction of the user's viewing trajectory is an important component of a practical interactive streaming system. In these experiments, two main questions are answered. First, how far ahead should the view trajectory be predicted? Second, what is the penalty in rate-distortion streaming performance incurred by using prediction over knowing the trajectory in advance perfectly?

In these experiments, the *Bust* dataset is used with the PRED4 encoding. All the three trajectory classes are used as well. When the predicted trajectory is used for scheduling instead of the actual view trajectory, as would occur in a practical system, increasing the amount of lookahead does not always help. Fig. 5 shows the rate-distortion streaming results for this scenario for the INTRA and PRED4 encodings and for all three trajectories.

There appears to be an optimal time horizon of prediction, depending on the type of trajectory. For instance, for the *slow* trajectory with its slow, deliberate motion, increasing the prediction lookahead beyond 200 ms does not improve performance, but it does not degrade performance either. For the *medium* trajectory with faster, less predictable motion, increasing the prediction lookahead beyond 200 ms actually degrades performance. This effect is even more pronounced for the *fast* trajectory with its rapid, erratic motion.

The reason why longer prediction lookahead does not necessarily help performance is because the prediction method described in Section III-B is not accurate for more than a few time

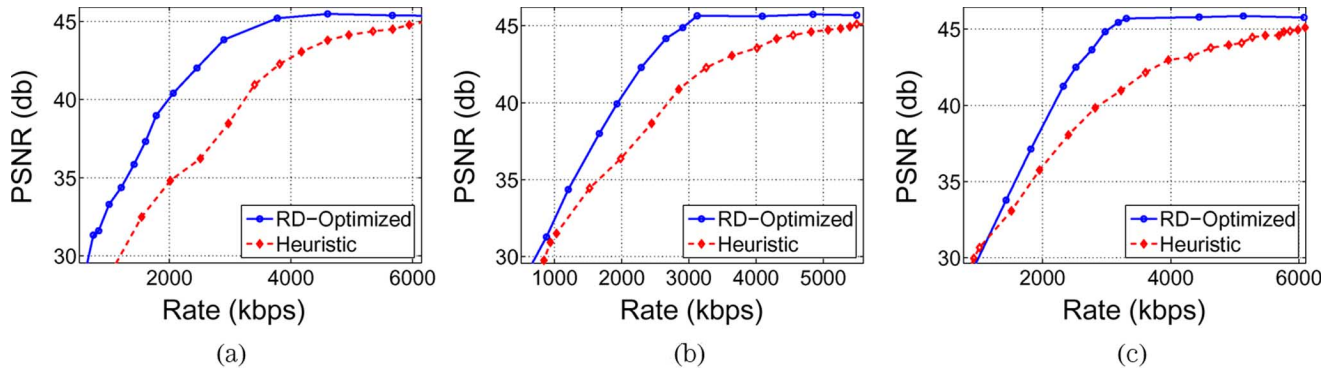

Fig. 3. Streaming results for the *Bust* light field, comparing rate-distortion optimized scheduling with heuristic scheduling. (a) INTRA—*medium* traj; (b) PRED2—*medium* traj; and (c) PRED4—*medium* traj.

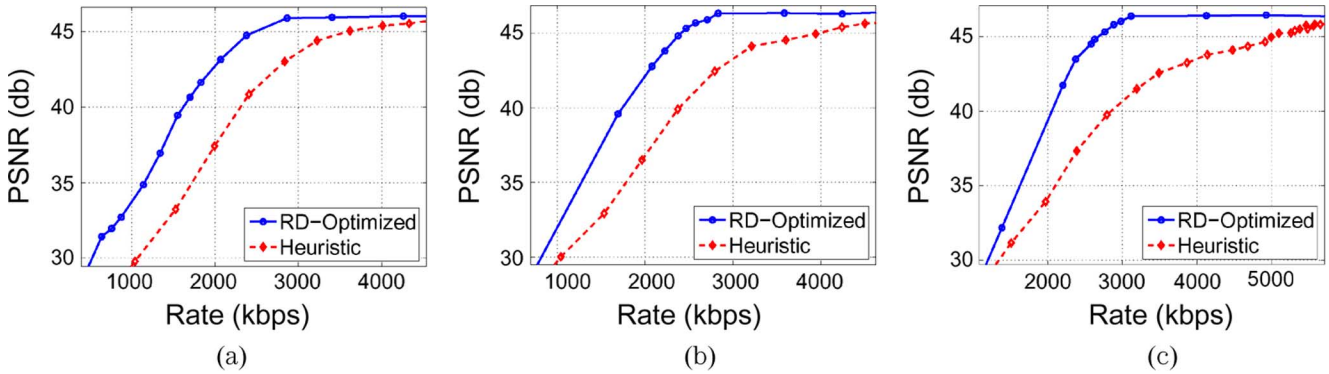

Fig. 4. Streaming results for the *Star* light field, comparing rate-distortion optimized scheduling with heuristic scheduling. (a) INTRA—*medium* traj; (b) PRED2—*medium* traj; and (c) PRED4—*medium* traj.

steps. If an inaccurate view trajectory is used, then image data that is not needed may be sent, decreasing the rate budget for images that are needed. It is interesting to note that in the very high bit-rate region in Fig. 5, looking further ahead does not hurt performance, since quality cannot be further improved and there is available rate to send images that may not be necessary.

The second question in this section is how the accuracy of view prediction affects performance. Fig. 6 compares the rate-distortion performance when using the actual trajectory versus using the predicted trajectory for scheduling. Results for the PRED4 encoding, and a prediction lookahead of 200 ms, are shown.

Since prediction is inaccurate, there is always a rate-distortion penalty for using the predicted view trajectory instead of the actual view trajectory. The penalty is, in general, larger for the more erratic faster view trajectories than for the slower. For the PRED4 encoding, there is a degradation in PSNR of 1 dB for the *slow* trajectory, and up to 2–3 dB for the other trajectories.

The experiments described in this section show that the performance of the streaming system can be greatly improved if it is possible to improve upon the view trajectory prediction. The rate-distortion penalties range from 1 dB to 3 dB in PSNR at the same rate for using the predicted trajectory in scheduling instead of the true trajectory. Since view trajectory prediction is inaccurate, there is an optimal point of prediction lookahead, in this case, 200 ms.

C. Encoding Prediction Dependency

The encoding scheme has strong influence on the compression efficiency as well as on the random access to individual images. This can affect streaming performance. Figs. 7 and 8 show the rate-distortion streaming performance for the *Bust* and *Star* light fields, respectively, and for all three trajectories.

An interesting observation is that using no prediction (INTRA) often does better than using 2 or more levels of prediction. This can be easily observed for the *Star* light field, in which INTRA coding gives the best streaming performance for a large range of bit rates, even for the *fast* trajectory. For the *Bust* light field, this can be observed at least for the *slow* trajectory.

One reason why this occurs is because using prediction between images tends to restrict random access. For instance, consider the scenario where the rate constraints dictate that only a few images can be transmitted for a particular view. When using prediction, these images are necessarily restricted to those in the lower levels of the prediction hierarchy. With independent coding of images, there is no such restriction and images that are closer to the desired view can be sent, resulting in lower distortion. Prediction, however, tends to decrease the size of the compressed images that are transmitted. In addition, the bit-rate cost of the images sent from the lower levels of the prediction hierarchy can be amortized over several views when considering

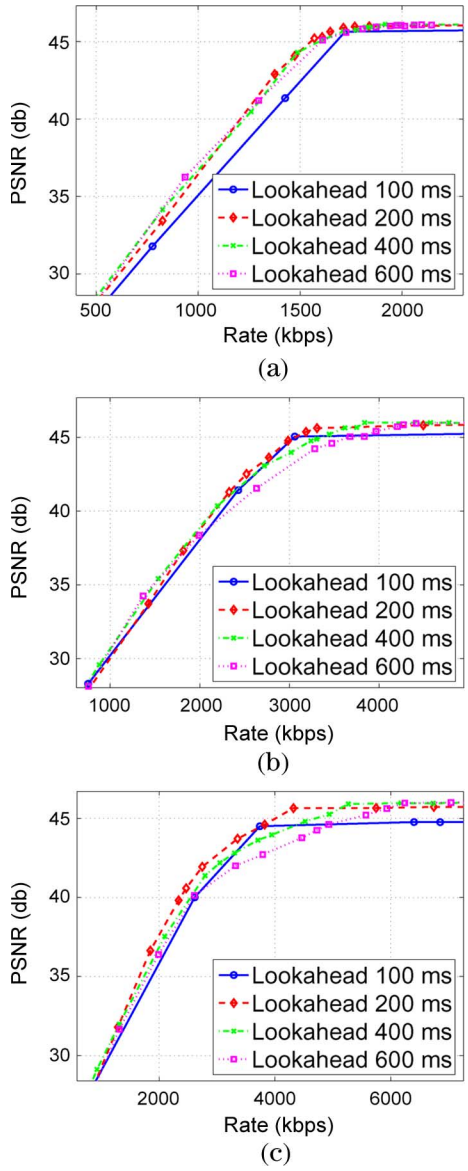

Fig. 5. Rate-distortion optimized streaming results for the *Bust* light field, with different amounts of prediction lookahead. The viewing trajectory is predicted. (a) PRED4—*slow* trajectory; (b) PRED4—*medium* trajectory; and (c) PRED4—*fast* trajectory.

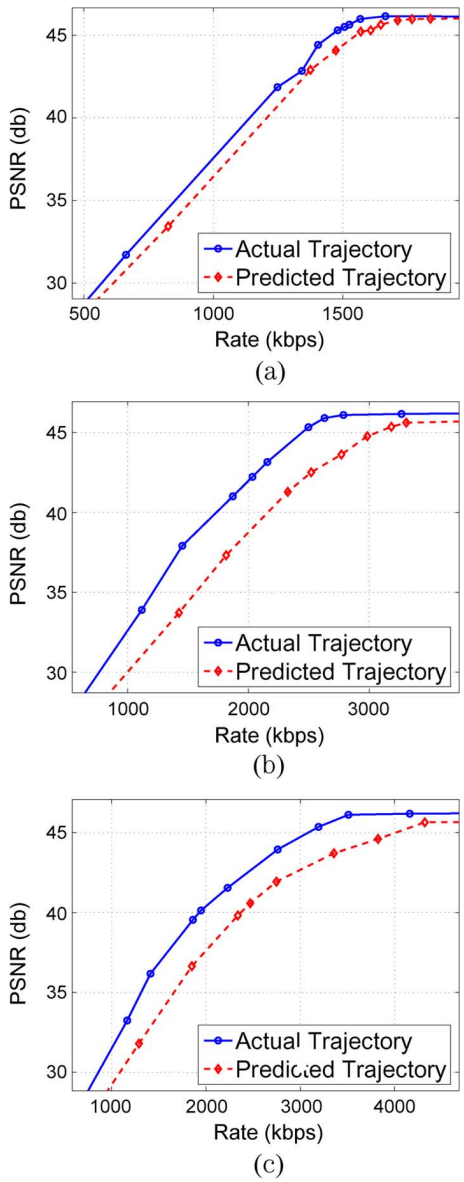

Fig. 6. Rate-distortion optimized streaming results for the *Bust* light field, comparing using the predicted trajectory versus the actual trajectory in the scheduling. The prediction lookahead is 200 ms. (a) PRED4—*slow* trajectory; (b) PRED4—*medium* trajectory; and (c) PRED4—*fast* trajectory.

several views in a trajectory. This tradeoff between decreased image size and decreased random access depends on the viewing trajectory.

As the trajectory goes from *slow* to *medium* to *fast*, more of the viewing hemisphere is covered. In general, this means that the total number of images required to render all the views in the trajectory increases. If several levels of prediction are used, this means that the overhead of sending the lower levels of the hierarchy are better amortized with a larger number of images. Thus, for the *fast* trajectory, with more images to be encoded, the overhead of additional prediction levels is better justified. The decrease in the size of the encoded images when using prediction more than compensates for this overhead, and leads to better streaming performance.

VI. Conclusion

This paper presents and investigates an interactive light field streaming system where a user remotely accesses a dataset over a best-effort packet network.

Interactivity imposes a stringent low-latency constraint on the system. In the system that is presented, prediction of the user's future viewing trajectory is used to mitigate the effects of this low-latency requirement. Experimental results show that there is an optimal horizon of prediction lookahead, which balances the benefits of looking ahead with the limited accuracy of predicting future user movements. The results also indicate that further improvements in rate-distortion streaming performance are possible if the view prediction scheme can be made more accurate.

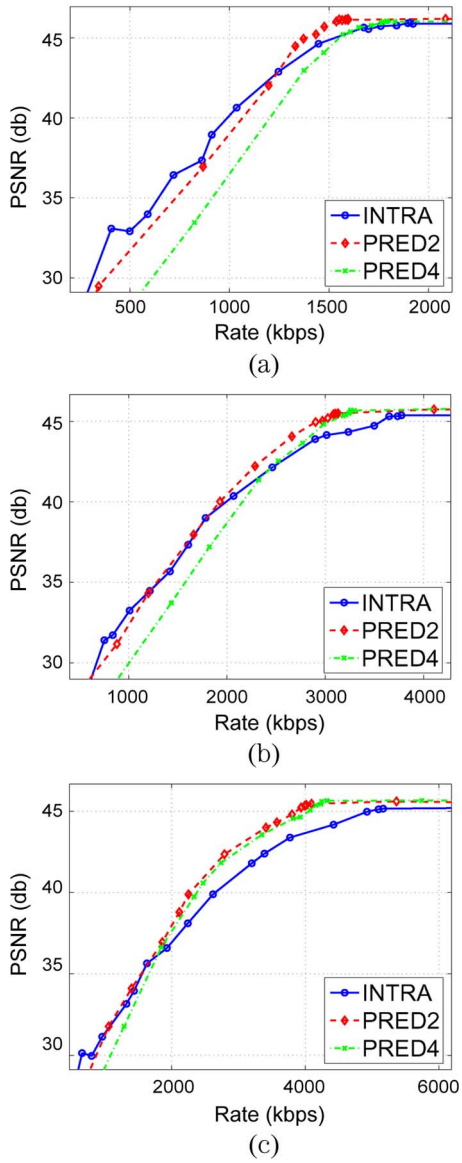

Fig. 7. Streaming results for the *Bust* light field, comparing three different encodings. Receiver-driven rate-distortion optimized streaming is used. (a) *slow* trajectory; (b) *medium* trajectory; and (c) *fast* trajectory

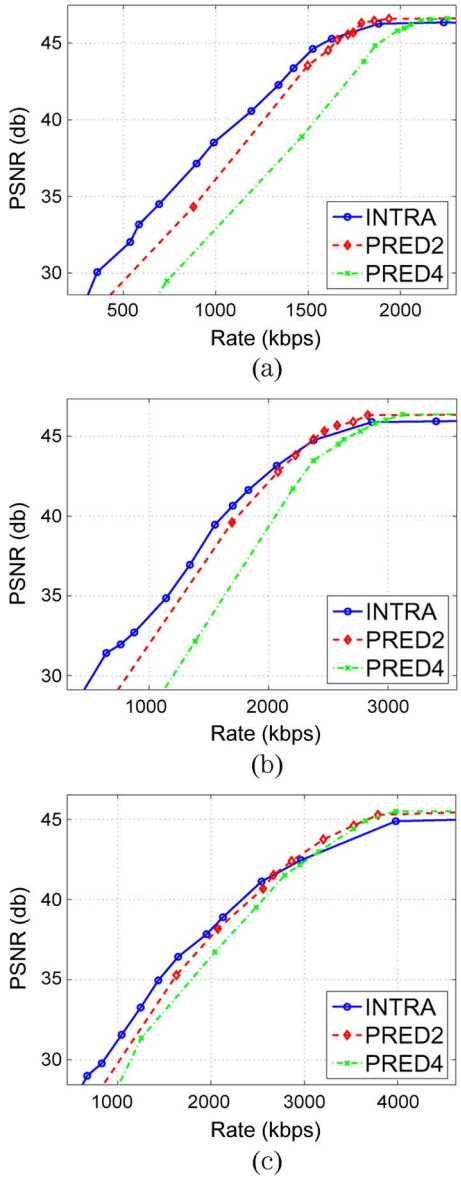

Fig. 8. Streaming results for the *Star* light field, comparing three different encodings. Receiver-driven rate-distortion optimized streaming is used. (a) *slow* trajectory; (b) *medium* trajectory; and (c) *fast* trajectory.

A key component of the light field streaming system is the packet scheduler which decides which images to transmit, and potentially re-transmit, to the remote user. A rate-distortion optimized packet scheduling framework is proposed. This framework takes into account network loss and delay statistics, the sizes of individual images, the distortion contribution of a set of images for a particular view, decoding dependencies between images, and knowledge of which images have been received or acknowledged. Experimental results show that rate-distortion optimized scheduling can outperform a heuristic scheduling algorithm by up to 2–5 dB in PSNR, at the same rate, depending on the dataset, user view trajectory, and average transmission rate. If the computationally complex task of scheduling is performed at the receiver instead of the sender, then it is possible to simultaneously serve a large number of clients.

For both rate-distortion optimized and heuristic scheduling, the encoding prediction structure has a significant effect on the streaming performance. While using more levels of prediction tends to improve compression efficiency, it also creates decoding dependencies that are detrimental to streaming performance. Experimental results show, for some datasets and trajectories, that independent encoding of the images provides better streaming performance than using more levels of prediction.

The light field streaming system currently assumes that only a few images are needed to render a novel view. While this is valid on the hemisphere of capturing cameras, many images are typically needed when either zooming in or out. For efficient streaming performance, the coding and streaming algorithms presented here will need to be adapted to this new scenario.

REFERENCES

[1] S. Chen, "Quicktime VR—an image-based approach to virtual environment navigation," in *Proc. SIGGRAPH95*, Aug. 1995, pp. 29–38.

[2] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. SIGGRAPH96*, Aug. 1996, pp. 31–42.

[3] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. SIGGRAPH96*, Aug. 1996, pp. 43–54.

[4] T. Fujii, "A Basic Study on the Integrated 3-D Visual Communication," Ph.D. dissertation, Univ. Tokyo, Tokyo, Japan, 1994.

[5] T. Fujii, T. Kimoto, and M. Tanimoto, "Ray space coding for 3-D visual communication," in *Proc. Picture Coding Symp. PCS-1996*, Melbourne, Australia, Mar. 1996, pp. 447–451.

[6] H.-Y. Shum and L.-W. He, "Rendering with concentric mosaics," in *Proc. SIGGRAPH99*, Aug. 1999, pp. 299–306.

[7] L. McMillan, "An Image-Based Approach to Three-Dimensional Computer Graphics," Ph.D. dissertation, Univ. North Carolina, Chapel Hill, NC, 1999.

[8] J. Shade, S. Gortler, L.-W. He, and R. Szeliski, "Layered depth images," in *Proc. SIGGRAPH98*, 1998, pp. 231–242, ACM Press.

[9] C.-F. Chang, G. Bishop, and A. Lastra, "LDI tree: A hierarchical representation for image-based rendering," in *Proc. SIGGRAPH99*, 1999, pp. 291–298.

[10] P. E. Debevec, C. J. Taylor, and J. Malik, "Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach," in *Proc. SIGGRAPH96*, 1996, pp. 11–20, ACM Press.

[11] P. E. Debevec, G. Borshukov, and Y. Yu, "Efficient view-dependent image-based rendering with projective texture-mapping," in *Proc. Eurographics Workship on Rendering*, Vienna, Austria, 1998, pp. 105–116.

[12] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, "Surface light fields for 3-D photography," in *Proc. SIGGRAPH 00*, Aug. 2000, pp. 287–296.

[13] M. Levoy and K. Pulli *et al.*, "The Digital Michelangelo project: 3-D scanning of large statues," in *Proc. SIGGRAPH 00*, Aug. 2000, pp. 131–144.

[14] M. Lukacs, "Predictive coding of multi-viewpoint image sets," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing ICASSP 1986*, 1986, pp. 521–524.

[15] I. Dinstein, G. Guy, J. Rabany, J. Tzelgov, and A. Henik, "On stereo image coding," in *Proc. Int. Conf. Pattern Recognition*, Nov. 1988, vol. 1, pp. 357–359.

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

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

[18] H. Aydinoğlu and M. H. H. , III, "Compression of multi-view images," in *Proc. IEEE Int. Conf. on Image Processing ICIP-1994*, 1994, vol. 2, pp. 385–389.

[19] H. Aydinoğlu, F. Kossentini, and M. H. H. , III, "A new framework for multi-view image coding," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing ICASSP-95*, 1995, vol. 4, pp. 2173–2176.

[20] 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.

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

[22] P. Ramanathan, M. Flierl, and B. Girod, "Multi-hypothesis disparity-compensated light field compression," in *Proc. IEEE Int. Conf. Image Processing ICIP-2001*, Oct. 2001.

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

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

[25] C.-L. Chang, X. Zhu, P. Ramanathan, and B. Girod, "Shape adaptation for light field compression," in *Proc. IEEE Int. Conf. Image Processing ICIP-2003*, Barcelona, Spain, Sep. 2003.

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

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

[28] C.-L. Chang, X. Zhu, P. Ramanathan, and B. Girod, "Inter-view wavelet compression of light fields with disparity-compensated lifting," in *Proc. SPIE Visual Comm. and Image Processing VCIP-2003*, Lugano, Switzerland, Jul. 2003.

[29] B. Girod, C.-L. Chang, P. Ramanathan, and X. Zhu, "Light field compression using disparity-compensated lifting," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing 2003*, Hong Kong, China, Apr. 2003, vol. IV, pp. 761–764.

[30] X. Tong and R. M. Gray, "Interactive rendering from compressed light fields," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 11, pp. 1080–1091, Nov. 2003.

[31] ITU-T, ISE/IEC 15444-1:2000, Information Technology: JPEG 2000 Image Coding System 2002.

[32] D. Taubman and R. Rosenbaum, "Rate-distortion optimized interactive browsing of JPEG2000 images," in *Proc. IEEE Int. Conf. Image Processing ICIP-2003*, Barcelona, Spain, Sep. 2003, vol. 2, pp. 765–768.

[33] D. S. Cruz, T. Ebrahimi, M. Larsson, J. Askelof, and C. Christopoulos, "Region of interest coding in JPEG2000 for interactive client/server applications," in *Proc. Multimedia Signal Processing Workshop MMSP-1999*, Copenhagen, Denmark, Sep. 1999, pp. 389–394.

[34] I. Peter and W. Strasser, "The wavelet stream: Interactive multi resolution light field rendering," in *Proc. 12th Eurographics Workshop on Rendering*, Jun. 2001, pp. 262–273.

[35] C.-L. Chang and B. Girod, "Rate-distortion optimized interactive streaming for scalable bitstreams of light fields," in *Proc. SPIE Visual Comm. and Image Processing VCIP-2004*, Jan. 2004.

[36] ——, "Receiver-based rate-distortion optimized interactive streaming for scalable bitstreams of light fields," in *Proc. IEEE Int. Conf. Multimedia Expo ICME-2004*, Jun. 2004.

[37] C. Zhang and J. Li, "Interactive browsing of 3-D environment over the Internet," in *Proc. SPIE Visual Comm. and Image Processing VCIP-2001*, San Jose, CA, Jan. 2001, pp. 509–520.

[38] ——, "Compression and rendering of concentric mosaics with reference block codec (RBC)," in *Proc. SPIE Visual Comm. and Image Processing VCIP-2000*, 2000, pp. 43–54.

[39] D. Gotz and K. Mayer-Patel, "A general framework for multidimensional adaptation," in *Proc. ACM Multimedia 2004*, 2004, pp. 612–619.

[40] ——, "A framework for scalable delivery of digitized spaces," *Int. J. Digital Libraries*, vol. 5, no. 3, pp. 205–218, May 2005.

[41] A. Albanese, J. Blömer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1737–1744, Nov. 1996.

[42] M. Khansari, A. Jalali, E. Dubois, and P. Mermelstein, "Low bit-rate video transmission over fading channels for wireless microcellular systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 1, pp. 1–11, Feb. 1996.

[43] H. Liu and M. E. Zarki, "Performance of H.263 video transmission over wireless channels using hybrid ARQ," *IEEE J. Select. Areas Commun.*, vol. 15, no. 9, pp. 1775–86, Dec. 1997.

[44] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, Apr. 2006.

[45] P. A. Chou and A. Seghal, "Rate-distortion optimized receiver-driven streaming over best-effort networks," in *Packet Video Workshop*, Pittsburgh, PA, Apr. 2002.

[46] ITU-T, One-Way Transmission Time ITU-T Recommend. G.114, Feb. 1996.

[47] R. Azuma and G. Bishop, "A frequency-domain analysis of head-motion prediction," in *Proc. SIGGRAPH95*, Aug. 1995, pp. 401–408.

[48] S. K. Singhal and D. R. Cheriton, "Exploiting position history for efficient remote rendering in networked virtual reality," *Presence: Teleoperators Virtual Environ.*, vol. 4, pp. 169–193, Spring 1995.

[49] P. Ramanathan, Light Field Viewer Software [Online]. Available: http://www.ivms.stanford.edu/pramanat/lfviewer.html 2003

[50] P. Ramanathan, M. Kalman, and B. Girod, "Rate-distortion optimized streaming of compressed light fields," in *Proc. IEEE Int. Conf. on Image Processing ICIP-2003*, Barcelona, Spain, Sep. 2003, vol. 3, pp. 277–280.

[51] P. Ramanathan and B. Girod, "Theoretical analysis of the rate-distortion performance of a light field streaming system," in *Proc. Picture Coding Symp. (PCS-2004)*, San Francisco, CA, Dec. 2004.

[52] ——, "Rate-distortion analysis of random access for compressed light fields," in *Proc. IEEE Int. Conf. Image Processing ICIP-2004*, Singapore, Oct. 2004.

[53] P. Ramanathan, "Compression and Interactive Streaming of Light Fields," Ph.D. dissertation, Stanford University, Stanford, CA, 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.

He has worked on various topics, including computer vision and graphics, image and video compression, and multimedia streaming, and has co-authored more than 20 journal and conference papers. From 2005 to 2006, he worked for an early-stage networking start-up company in Santa Clara, CA, on efficient compression and transmission for a remote desktop system. He is now with Truveo, part of AOL, LLC, San Francisco, CA, working on internet video search.

Mark Kalman received the B.S. degree in electrical engineering and the B.Mus. degree in composition from Johns Hopkins University, Baltimore, MD, in 1997. He received the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, in 2001 and 2006, respectively.

He is currently with Pure Digital Technologies, Inc., San Francisco, CA.

Bernd Girod (F'98) received the M.S. degree in electrical engineering from Georgia Institute of Technology, Atlanta, in 1980 and the Doctoral degree "with highest honours" from the University of Hannover, Germany, in 1987.

He is Professor of electrical engineering in the Information Systems Laboratory of Stanford University, Stanford, CA. He also holds a courtesy appointment with the Stanford Department of Computer Science. He serves as Director both of the Stanford Center for Image Systems Engineering (SCIEN) and the Max Planck Center for Visual Computing and Communication. Since 2004, he also has served as Chairman of the Steering Committee of the new Deutsche Telekom Laboratories at the Technical University of Berlin. His research interests include video coding and networked media systems. Until 1987, he was a Member of Research staff at the Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, University of Hannover. In 1988, he joined Massachusetts Institute of Technology, Cambridge, MA, USA, first as a Visiting Scientist with the Research Laboratory of Electronics, then as an Assistant Professor of Media Technology at the Media Laboratory. From 1990 to 1993, he was Professor of Computer Graphics and Technical Director of the Academy of Media Arts in Cologne, Germany, jointly appointed with the Computer Science Section of Cologne University. He was a Visiting Adjunct Professor with the Digital Signal Processing Group at Georgia Institute of Technology, Atlanta, GA, USA, in 1993. From 1993 until 1999, he was Chaired Professor of Electrical Engineering/Telecommunications at University of Erlangen-Nuremberg, Germany, and the Head of the Telecommunications Institute I, co-directing the Telecommunications Laboratory. He served as the Chairman of the Electrical Engineering Department from 1995 to 1997, and as Director of the Center of Excellence "3-D Image Analysis and Synthesis" from 1995–1999. He was a Visiting Professor with the Information Systems Laboratory of Stanford University during the 1997/98 academic year. As an entrepreneur, he has worked successfully with several start-up ventures as founder, investor, director, or advisor. Most notably, he has been a co-founder and Chief Scientist of Vivo Software, Inc., Waltham, MA (1993–1998); after Vivo's aquisition, 1998–2002, Chief Scientist of RealNetworks, Inc. He has served on the Board of Directors for 8 × 8, Inc., Santa Clara, CA, 1996–2004, and for GeoVantage, Inc., Swampscott, MA, 2000–2005. He is currently an advisor to start-up companies Mobilygen, Santa Clara, CA, and to NetEnrich, Inc., Santa Clara, CA. He has authored or co-authored one major text-book (printed in three languages), three monographs, and over 350 book chapters, journal articles, and conference papers, and he holds over 20 U.S. patents.

Dr. Girod has served as on the editorial boards for several journals in his field, among them as Area Editor for Speech, Image, Video, and Signal Processing for the IEEE TRANSACTIONS ON COMMUNICATIONS. He has served on numerous conference committees, e.g., as Tutorial Chair of ICASSP-97 in Munich and again for ICIP-2000 in Vancouver, as General Chair of the 1998 IEEE Image and Multidimensional Signal Processing Workshop in Alpbach, Austria, as General Chair of the Visual Communication and Image Processing Conference (VCIP) in San Jose, CA, in 2001, and General Chair of Vision, Modeling, and Visualization (VMV) at Stanford, CA, in 2004. He was a member of the IEEE Image and Multidimensional Signal Processing Technical Committee from 1989 to 1997. He was elected Fellow of the IEEE in 1998 "for his contributions to the theory and practice of video communications." He was named Distinguished Lecturer for the year 2002 by the IEEE Signal Processing Society. He received the 2002 EURASIP Best Paper Award (with J. Eggers) and the 2004 EURASIP Techical Achievement Award.