JPEG2000-Based Scalable Interactive Video (JSIV) with Motion Compensation

Aous Thabit Naman, *Member, IEEE,* and David Taubman, *Senior Member, IEEE*

***Abstract*—In a recent work, the authors proposed a novel paradigm for interactive video streaming and coined the term JPEG2000-Based Scalable Interactive Video (JSIV) for it. In this work, we investigate JSIV when motion compensation is employed to improve prediction, something that was intentionally left out in our earlier treatment. JSIV relies on three concepts: storing the video sequence as independent JPEG2000 frames to provide quality and spatial resolution scalability; prediction and conditional replenishment of code-blocks to exploit inter-frame redundancy; and loosely-coupled server and client policies in which a server optimally selects the number of quality layers for each code-block transmitted and a client makes the most of the received (distorted) frames. In JSIV, the server transmission problem is optimally solved using Lagrangian-style rate-distortion optimization. The flexibility of JSIV enables us to employ a wide variety of frame prediction arrangements, including hierarchical B-frames. JSIV provides considerably better interactivity compared to existing schemes and can adapt immediately to interactive changes in client interests, such as forward or backward playback and zooming into individual frames. Experimental results show that JSIV's performance is inferior to that of SVC in conventional streaming applications while JSIV performs better in interactive browsing applications.**

***Index Terms*—Teleconferencing, video signal processing, image coding, image communication, weighted acyclic directed graphs.**

I. Introduction

REMOTE interactive browsing of video has traditionally been limited to pause and random access to some predetermined access points. This limitation is a result of employing standard video compression techniques that can, at best, provide limited interactivity options.

This limited interactivity and other issues motivated research in scalable video coding that can provide considerably better interactivity and can solve some of the other existing problems in video storage and streaming. Research in the area has produced some promising results [1], [2] and recently a scalable video coding (SVC) extension to H.264/AVC [3] has been approved within the ISO working group known as MPEG, to provide improved scalability options.

Despite these improved options, the encoder still imposes restrictions on the encoded stream that limit accessibility. For example, in order to deliver a given frame to the client, a server would have to send enough data from the group of pictures (GOP) that contains this frame, possibly the whole GOP, and the client would have to decode potentially many frames in order to invert the motion compensated transform used during compression and extract the desired frame. The reader can find other examples where accessibility is limited by the encoder in [4]–[6].

The JPEG2000 Interactive Protocol (JPIP) [7], [8], on the other hand, provides many of the desirable features of scalable interactive browsing, such as spatial and temporal scalability, quality scalability, and spatial and temporal accessibility, but it only works for still images and motion-JPEG2000[1].

Recently, we proposed the JPEG2000-Based Scalable Interactive Video (JSIV) paradigm as a way to provide better flexibility, scalability, and interactivity for video streaming and browsing. Some of JSIV's concepts were introduced progressively in [6], [10]–[14]; the objective of this work and the work in [5] is to elaborate on the concepts behind JSIV and formalize them.

In [5], we discussed the philosophy behind JSIV and the scenarios in which JSIV is favorable (the interested reader can refer to [4], [5] for more details), but the discussions in [5] were intentionally limited to the case in which JSIV employs prediction without motion compensation. In this work, we turn our focus to the case of JSIV with motion compensation. To this end, we investigate the effect of motion compensation on distortion propagation from reference frames to predicted frames, we propose a way of approximating distortions when motion compensation is involved, and we investigate the accuracy of these approximations and their storage requirements and computational costs. We also propose policies that facilitate a realistic implementation of a JSIV system that employs motion compensation.

JSIV relies on:

- JPEG2000 to independently compress the individual frames of the video sequence and provide for quality and spatial resolution scalability as well as random accessibility.
- Prediction, with or without motion compensation, and conditional replenishment of JPEG2000 code-blocks to exploit temporal redundancy.
- Loosely-coupled server and client policies. The server policy aims to select the best number of quality layers for each precinct it serves and the client policy attempts to produce the best possible reconstructed frames from the data the client has. Each of these policies may evolve separately without breaking the communication paradigm.

Figure 1 shows a simplified block diagram of a JSIV system that employs motion compensation. The system has three basic

Copyright ©2010 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org

[1]Motion-JPEG2000 [9] is a video file format based on JPEG2000. The file contains some video timing information, and each frame is stored independently in its own code-stream.

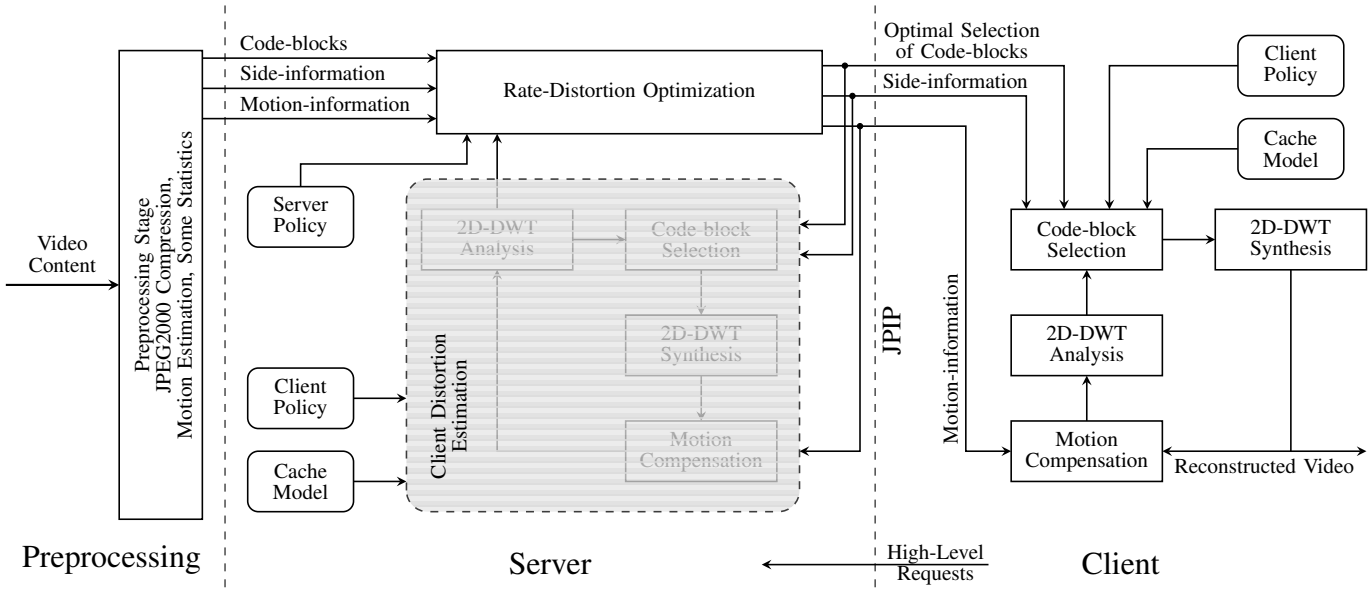

Fig. 1. A simplified block diagram of the proposed JSIV delivery system. The client distortion estimation block, shown in gray, estimates client-side distortions in reconstructed frames without reconstructing them.

entities: the preprocessing stage, the server, and the client. The preprocessing stage is responsible for compressing each frame individually into JPEG2000 format and preparing the side information needed during video serving.

The server is composed of two main sub-blocks: the client distortion estimation block (CDEB) and the rate-distortion optimization block (RDOB). The CDEB models the distortions in each code-block or precinct of each frame, taking into consideration the effect of motion compensation at the client, and generating approximate distortion information without actually reconstructing any frames; it relies on its knowledge about transmitted information and an assumed client policy which does not need to be exact. The RDOB performs Lagrangian-style rate-distortion optimization to decide the number of quality layers to be sent for each precinct of each frame. It also decides on any side information (including motion information) needed by the client to best exploit the frame data.

The server communicates with the client employing only JPIP [7], [8]; JSIV stores side information in additional components in each frame, conceptually known as meta-components or meta-images. This allows the use of JPIP without any modifications to send both code-block data and side information.

The client receives compressed code-block bit-streams and side information (including motion information). Using this information and aided by a client policy, the client selects the source of data to use for each code-block. In particular, the client has the option to decode an available code-block bit-stream directly or to predict the code-block from nearby frames (possibly having much higher quality), employing motion compensation.

A number of researchers have realized the limited interactivity provided by conventional video streaming practices [15]–[20], and have devised different approaches that are favorable in certain situations. A survey of these approaches is given in [4], [5].

The rest of this work is organized as follows. Section II discusses the effects of motion compensation on distortion propagation. Section III describes "oracle" client and server policies that enable us to discuss the basic JSIV optimization algorithm. Section IV proposes a way of significantly reducing the computational cost associated with estimating distortions within the server. Section V gives the actual client and server policies and elaborates on side-information delivery. In Section VI, we discuss the computational cost and storage requirements for JSIV deployment. Section VII gives some experimental results which allow JSIV to be compared with traditional video coding approaches. Section VIII discusses the effects of the approximations we introduced in order to achieve a realistic implementation of JSIV. Finally, Section IX states our conclusions and points to future work.

II. The Effects of Motion Compensation on Distortion Propagation

The use of motion compensation to improve prediction is central to this paper, and we find it convenient at this stage to consider the effect of motion compensation on distortion propagation from reference frames to predicted frames. We present a quantitative analysis of this effect in Section IV.

JSIV employs JPEG2000 to store individual frames; JPEG2000 utilizes the two-dimensional discrete wavelet transform (2D-DWT) to decompose a frame, f_n, into a set of sub-bands, and each sub-band is partitioned into rectangular blocks known as *code-blocks*, $\mathcal{C}_n^\beta$, as shown in Figure 2. Although code-blocks are coded independently, they are not explicitly identified within the code-stream; code-blocks are collected into larger groupings known as *precincts*, $\mathcal{P}_n^\pi$, also shown

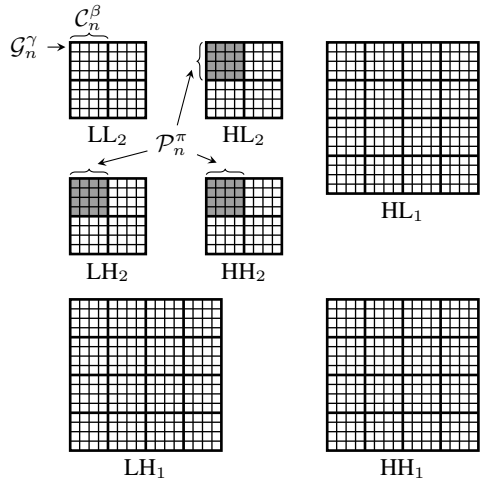

Fig. 2. Relation between the different partitions in this work. 2D-DWT decomposes a given frame, f_n, into sub-bands; a two-level decomposition is shown with sub-band labels that follow sub-band naming conventions. Each sub-band is partitioned into code-blocks, $\mathcal{C}_n^\beta$; in this figure, for example, each sub-band at the lower decomposition level, LL_2, HL_2, LH_2, and HH_2, has 4 code-blocks while, at the higher level, each has 16 code-blocks. Each sub-band is also partitioned into smaller blocks, known as grid blocks. A grid block, $\mathcal{G}_n^\gamma$, is shown as a small square; in the figure, each code-block has 16 grid blocks. A precinct, $\mathcal{P}_n^\pi$, groups code-blocks that contribute to the same spatial region from three sub-bands, HL_d, LH_d, and HH_d, at a given decomposition level, d; precincts for the LL_D sub-band, where D is the number of decomposition level, contains code-blocks from that sub-band only.

in Figure 2. For image browsing/streaming applications it is preferable that each precinct has only one code-block from each of its constituent sub-bands since this minimizes the spatial impact of a precinct.

In JSIV, the samples of a code-block are obtained either from decoding the zero or more quality layers, q_n^β, available for that code-block or by predicting them from nearby frames; we write $\mathcal{C}_{*n}^\beta$ for the de-quantized samples and $\mathcal{C}_{\to n}^\beta$ for the predicted samples.

In this work, prediction involves the use of motion compensation; we always employ motion compensation to synthesized frames at the highest available resolution[2]. A widely-employed technique for improving prediction in predictive video coding schemes is to use some position-dependent linear combination of more than one predictor; each of these predictors is obtained from some reference frame using motion compensation. Here, we write $\mathcal{A}(f_n)$ for the set of reference frames that directly contribute to f_n's prediction, and we employ a linear combination given by

$$f_{\to n} = \sum_{f_r \in \mathcal{A}(f_n)} g_{rn} \cdot \mathcal{W}_{r \to n}(f_r) \tag{1}$$

where $\mathcal{W}_{a \to b}$ is the motion compensation operator mapping f_a to f_b. We choose to use position-independent scaling factors, g_{rn}, in this work; space-varying scaling factors, however, can be readily incorporated into the approach. Thus, predicted samples of a given code-block, $\mathcal{C}_{\to n}^\beta$, are obtained by applying the 2D-DWT to $f_{\to n}$ and selecting the appropriate sub-band and region that corresponds to $\mathcal{C}_n^\beta$.

Rather than estimating distortions on a code-block basis, we estimate them on a finer grid; we partition each sub-band in frame f_n into rectangular blocks that we name *grid blocks* and denote by $\mathcal{G}_n^\gamma$, as shown in Figure 2. The reason for this finer partitioning is to provide a finer description of distortion in the event that a predicted frame becomes itself a reference frame for motion compensation; this case is depicted in Figure 3 where frame f_i is directly decoded (i.e. decoded independently), frame f_j is predicted from f_i, and frame f_k is predicted f_j. We discuss the effect of grid block dimensions on the accuracy of distortion modeling in Section VIII. In summary, each precinct, $\mathcal{P}_n^\pi$, contains one or more code-blocks, $\mathcal{C}_n^\beta$; each of which contains one or more grid blocks $\mathcal{G}_n^\gamma$.

We write $D_{*n}^\gamma = \|\mathcal{G}_{*n}^\gamma - \mathring{\mathcal{G}}_n^\gamma\|^2$ for the distortion associated with de-quantized samples of grid block $\mathcal{G}_n^\gamma$ in frame f_n, where $\mathring{\mathcal{G}}_n^\gamma$ represent the full-quality grid block samples. Similarly, we write $D_{\to n}^\gamma$ for the distortion associated with $\mathcal{G}_{\to n}^\gamma$. Using an additive distortion model, the frame distortion attributed to a precinct $\mathcal{P}_n^\pi$ can be approximated by

$$D_n^\pi = \sum_{\mathcal{G}_n^\gamma \subset \mathcal{P}_n^\pi} G_{b_\gamma} \cdot D_n^\gamma \tag{2}$$

where G_{b_γ} is the energy gain factor of sub-band b_γ to which grid block $\mathcal{G}_n^\gamma$ belongs and $\mathcal{G}_n^\gamma \subset \mathcal{P}_n^\pi$ enumerates the grid blocks contained within precinct $\mathcal{P}_n^\pi$. Similar approximations can also be written for both D_{*n}^π and $D_{\to n}^\pi$. These precinct distortion approximations are valid provided that the wavelet transform basis functions are orthogonal or the quantization errors in each of the samples are uncorrelated. Neither of these requirements is strictly satisfied; however, the well-known CDF 9/7 wavelet kernels used in our experimental investigations in Section VII have nearly orthogonal basis functions.

Due to the shift-variant behavior and the slow response roll-off of the DWT, any distortion in grid block $\mathcal{G}_i^\gamma$ of frame f_i contributes, in general, to the distortion of more than one grid block in frame f_j when motion compensated prediction is employed; this behavior is depicted in Figure 3. We say that the distortion in $\mathcal{G}_i^\gamma$ *leaks* to the set of grid blocks denoted by $\mathcal{S}(\mathcal{G}_i^\gamma)$; this leakage behavior is also documented in [22].

We show in Section IV that the distortion energy which leaks from grid block $\mathcal{G}_i^\gamma$ of sub-band b_i in frame f_i to grid block $\mathcal{G}_j^\gamma$ of sub-band b_j in frame f_j can be approximated by $G_{\mathcal{W}(\gamma_j)}^{b_i \to b_j} \cdot D_i^\gamma$, where $G_{\mathcal{W}(\gamma_j)}^{b_i \to b_j}$ is a position-dependent (i.e. grid block dependent) *distortion gain*; the subscript of the distortion gain, $\mathcal{W}(\gamma_j)$, emphasizes the dependency of the distortion gain on the motion vector field around its respective grid block, $\mathcal{G}_j^\gamma$.

It is obvious from Figure 3 that prediction creates dependency among grid blocks of different frames; this dependency can be represented by a weighted acyclic directed graph (WADG) [23], as shown in Figure 4. In the context of Figure 4, the nodes of the graph represent grid-blocks, but, in general, they can represent frames, precincts, or code-blocks in other contexts. The *Antecedents* of node n, denoted by $\mathcal{A}(n)$ are the set of nodes that contribute to node n, and the *Succedents*

[2]An alternate approach is to employ in-band motion-compensation [21]; however, experimental results reveal that this choice has a negative impact on the quality of reconstructed video.

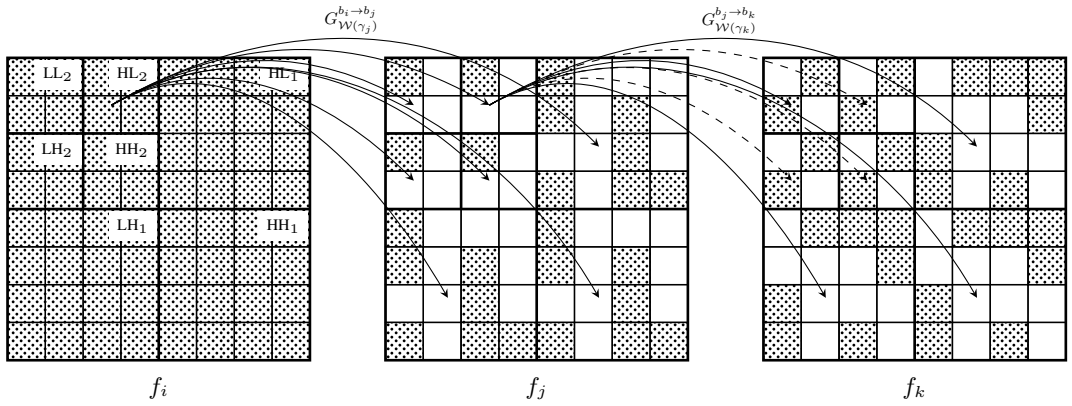

Fig. 3. Distortion propagation from reference frames to predicted frames; here, frame f_i is directly decoded, frame f_j is at least partially predicted from f_i, and frame f_k is at least partially predicted predicted from f_j. The figure shows a two-level decomposition of each frame with sub-band labels that follow sub-band naming conventions. The figure also shows code-blocks as squares; grid-blocks are not shown to reduce clutter. Decoded code-blocks are shown as (⊡), predicted code-blocks as (□). Arrows between f_i and f_j show distortion propagation from a given grid-block, $\mathcal{G}_i^\gamma$, to many grid blocks in f_j; we approximate the distortion propagated along each arrow by $G^{b_i \to b_j}_{\mathcal{W}(\gamma_j)}$ multiplied by the distortion in $\mathcal{G}_i^\gamma$. The dashed arrows between f_j and f_k represent possible distortion propagation that does not occur because the destination code-blocks in f_k are replaced by directly decoded code-blocks.

of node n, denoted by $\mathcal{S}(n)$ are the set of nodes that node n contributes to. The dependency graph is directed, with arcs, each of which is emanating from a grid block in a reference frame and ending in a grid block in a predicted frame; the weight along each arc represents the distortion gain, $G^{b_i \to b_j}_{\mathcal{W}(\gamma_j)}$. The dependency graph is acyclic because if $\mathcal{G}_i^\gamma$ is contributing to the prediction of $\mathcal{G}_j^\gamma$, there is no way, direct or indirect, that $\mathcal{G}_j^\gamma$ contributes to the prediction of $\mathcal{G}_i^\gamma$.

In general, the distortion associated with predicted samples, $D^\gamma_{\to n}$, is due to a combination of motion modeling errors (referred to as *motion distortion*) and errors in their prediction reference samples (referred to as *attributed distortion*); the errors in the reference samples are either due to quantization distortion (for decoded reference blocks) or a further combination of motion and attributed distortions (for predicted reference blocks). For example, grid block $\mathcal{G}_k^1$ in Figure 4a suffers from a combination of motion distortion and attributed distortions due to distortions in $\mathcal{A}(\mathcal{G}_k^1) = \{\mathcal{G}_j^2, \mathcal{G}_j^4\}$; grid block $\mathcal{G}_j^4$ suffers from quantization distortion while $\mathcal{G}_j^2$ suffers from other motion and attributed distortions.

In this work, we assume that motion and attributed error components are approximately uncorrelated, so that their squared error distortions are additive. Note, however, that the attributed distortion in $\mathcal{G}^\gamma_{\to n}$ may involve a mixture of quantization and other motion distortions depending on how each of its prediction source grid blocks was formed. This adds doubt about the validity of our assumption, since it requires successive motion distortions to be uncorrelated. Indeed, experimental results from [4], [5] and in Section VIII reveal that inaccuracies in this approximation have a measurable negative impact on the quality of reconstructed video, mainly due to accumulation of errors in estimating motion distortion. Nevertheless, we find this approximation necessary to develop a workable distortion estimation algorithm, as detailed in Subsection IV-A. Under this assumption, we can use (1) to write

$$D^\gamma_{\to n} \approx D^{\mathrm{M},\gamma}_{\to n} + \underbrace{\sum_{r \ni f_r \in \mathcal{A}(f_n)} g_{rn}^2 \cdot D^{\mathrm{A},\gamma}_{r \to n}}_{D^{\mathrm{A},\gamma}_{\to n}} \tag{3}$$

where $D^{\mathrm{M},\gamma}_{\to n}$ is the motion distortion and $D^{\mathrm{A},\gamma}_{\to n}$ is the attributed distortion. In the above, we have also assumed that errors among the different reference sources in $\mathcal{A}(f_n)$ are approximately uncorrelated, so that their squared error distortions add. The motion distortion is the distortion in grid block $\mathcal{G}_n^\gamma$ when full quality reference frames are used.

In the following paragraphs, we present a couple of examples to make this additive model clearer.

Example 1: Consider the distortion in $\mathcal{G}_k^1$ of Figure 4a. This distortion is equal to $D^{\mathrm{M},1}_{\to k} + D^{\mathrm{A},1}_{\to k}$, but $D^{\mathrm{A},1}_{\to k} = G^{2_j \to 1_k}_{\mathcal{W}(1_k)} \cdot D^4_{*j} + G^{1_j \to 1_k}_{\mathcal{W}(1_k)} \cdot D^2_{\to j}$. We also have $D^2_{\to j} = D^{\mathrm{M},2}_{\to j} + D^{\mathrm{A},2}_{\to j}$, where $D^{\mathrm{A},2}_{\to j} = G^{2_i \to 1_j}_{\mathcal{W}(2_j)} \cdot D^5_{*i} + G^{1_i \to 1_j}_{\mathcal{W}(2_j)} \cdot D^1_{*i}$; thus, $D^{\mathrm{A},1}_{\to k}$ can be written in terms of $D^{\mathrm{M},2}_{\to j}$ and $D^4_{*j}, D^5_{*i}, D^1_{*i}$. In fact, the distortion in any predicted grid block can be written as a linear combination of quantization distortion and motion distortion.

Another result of employing prediction is that the distortion in a given grid block, $\mathcal{G}_n^\gamma$, contributes distortion to all grid blocks in its succedents; that is, all the grid blocks in $\mathcal{S}(\mathcal{G}_n^\gamma)$ and in $\mathcal{S}(\mathcal{S}(\mathcal{G}_n^\gamma))$ and so forth, as shown in Figure 4a. It is useful to collect the contributions of grid block $\mathcal{G}_n^\gamma$ to the overall distortion of all frames under consideration in the form $(1 + \theta_n^\gamma) \cdot D_n^\gamma$, where θ_n^γ is what we call the *additional contribution weight*; these weights can be determined by traversing the converse of the dependency WADG as shown in Figure 4b.

Example 2: Consider the distortion contribution of grid block $\mathcal{G}_j^2$ to the distortion in frames f_j and f_k of Figure 4a. Grid block $\mathcal{G}_j^2$ contributed $G_1 \cdot D_j^2$ to frame f_j and $G_1 \cdot G^{1_j \to 1_k}_{\mathcal{W}(1_k)} \cdot D_j^2 + G_2 \cdot G^{1_j \to 2_k}_{\mathcal{W}(5_k)} \cdot D_j^2$, where G_1 and G_2 are the energy gain factors of sub-bands 1 and 2, respectively. The total contribution of grid block $\mathcal{G}_j^2$ can be written as $(1 + \theta_j^2) \cdot G_1 \cdot D_j^2$, where

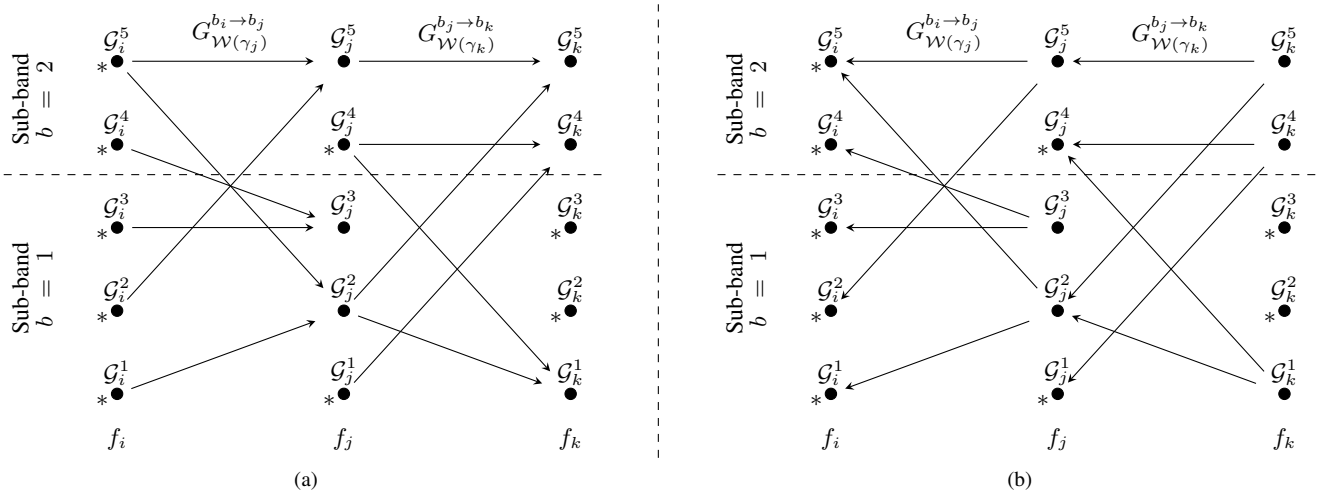

Fig. 4. (a) A typical WADG representing distortion propagation from reference grid blocks to predicted grid blocks. Each column represent one frame; frame f_i is directly decoded, frame f_j is predicted from f_i, and frame f_k is predicted from f_j. Each node represents one grid block; an "$*$" on the bottom-left side of a node indicates that the node is directly decoded rather than predicted. Each arrow represent distortion propagation with a distortion gain of $G^{b_i \to b_j}_{\mathcal{W}(\gamma_j)}$. (b) The converse of the WADG in (a). Arrows indicate back-propagation of contribution weights from predicted frames to reference frames.

$\theta^2_j = G^{1_j \to 1_k}_{\mathcal{W}(1_k)} + \frac{G_2}{G_1} \cdot G^{1_j \to 2_k}_{\mathcal{W}(5_k)}$; that is, in general, a grid block, $\mathcal{G}^\gamma_j$, contributes $(1+\theta^\gamma_j) \cdot G_{b_\gamma} \cdot D^\gamma_{\to j}$ when it is predicted and $(1+\theta^\gamma_j) \cdot G_{b_\gamma} \cdot D^\gamma_{*j}$ when it is directly decoded. Note that, when $\mathcal{G}^2_j$ is predicted, its distortion contribution can be written as $(1+\theta^2_j) \cdot G_1 \cdot D^{\mathrm{M},2}_{\to j} + (1+\theta^2_j) \cdot G_1 \cdot D^{A,2}_{\to j}$, where the last term represents contributions from grid blocks in f_i.

III. Oracle Client and Server Policies

In this work, prediction involves motion compensation at the client. The server itself calculates or estimates the impact of motion compensation so as to determine what content should be delivered, but the content which is delivered corresponds to independently compressed frames. The policies presented here are termed "oracle" policies because of the underlying unrealistic assumption that the client can replicate the server's rate-distortion optimization decisions, achieving the same quality of reconstructed video as that which the server is maximizing.

A. Oracle Client Policy

It is possible for the client to make decisions on a grid block basis, a code-block basis, or at the coarser level of precincts. We choose to work with precincts because the smallest piece a server can send in JPIP is one quality layer of one precinct (formally known as a packet); this means that the server transmits precinct-optimized data. Thus, for each precinct, $\mathcal{P}^\pi_n$, the client chooses either to use the received zero or more quality layers, q^π_n, that produce de-quantized samples, $\mathcal{P}^\pi_{*n}$, with an associated distortion of D^π_{*n} or to use predicted samples, $\mathcal{P}^\pi_{\to n}$, with an associated distortion of $D^\pi_{\to n}$. Ideally, the client chooses the samples that produce lower distortion; that is,

$$D^\pi_n = \min\{D^\pi_{*n}, D^\pi_{\to n}\} \tag{4}$$

This simple client policy is unrealistic, as the client has no access to the actual media and therefore is incapable of calculating distortions, especially for $D^\pi_{\to n}$; this policy will be revised in Section V with a realistic policy.

B. Oracle Server Policy

JSIV optimization is performed over windows of frames. Each frame within the window of frames (WOF) has a chance of contributing data to the interactive session. We refrain from using the term *group of pictures* (GOP) to describe these frames so as to avoid confusion; the selection of a WOF does not imply any particular predictive relationship between its frames.

The objective of the optimization problem at the server is to achieve the minimum possible distortion in the WOF, $\mathcal{F}_s$, being optimized, subject to some length constraint. Thus, the optimization problem involves selecting a number of quality layers, q^π_n, with an associated cost of $|q^\pi_n|$ bytes for each precinct, $\mathcal{P}^\pi_n$, in $\mathcal{F}_s$. Using (2), the server optimization problem is cast as the minimization of a cost functional, J_λ, given by

$$J_\lambda = \sum_{n \in \mathcal{F}_s} \sum_{\pi \in f_n} \sum_{\mathcal{G}^\gamma_n \subset \mathcal{P}^\pi_n} G_{b_\gamma} D^\gamma_n + \lambda \cdot \sum_{n \in \mathcal{F}_s} \sum_{\pi \in f_n} |q^\pi_n| \tag{5}$$

where λ is a Lagrangian parameter that is adjusted until the solution which minimizes J_λ satisfies the length constraint. The term that accounts for the cost associated with motion information is omitted from (5) because, currently, we do not employ a motion model that allows us, at serve-time and from compressed description, to trade accuracy of the motion model (distortion) for data rate on a *per code-block or precinct basis*. The motion model used here is expressed, formulated, and transmitted for frames as a whole rather than based on regions. For such a case, the motion information cost is a constant and can be ignored during optimization.

In general, each $\mathcal{F}_s$ has some of its precincts predicted, with distortion $D^\pi_{\to n}$, and some directly decoded, with distortion D^π_{*n}. We attach a hidden state variable, χ^π_n, to each precinct $\mathcal{P}^\pi_n$, where $\chi^\pi_n = 0$ for a predicted precinct and $\chi^\pi_n = 1$ for an decoded precinct. In practice, we perform all our distortion calculations on grid blocks $\mathcal{G}^\gamma_n$, but decisions on the number of quality layers q^π_n, and the state χ^π_n are still made on a precinct

basis. To stress this fact, we write $q_n^{\pi(\gamma)}$ for the number of quality layers associated with grid block $\mathcal{G}_n^\gamma$ such that this variable takes on the value of q_n^π associated with precinct $\mathcal{P}_n^\pi$ to which grid block $\mathcal{G}_n^\gamma$ belongs; that is $q_n^{\pi(\gamma)} = q_n^\pi$ for all $\mathcal{G}_n^\gamma \subset \mathcal{P}_n^\pi$. This way, (5) can be written as

$$\begin{aligned} J_\lambda = &\sum_{n\in\mathcal{F}_s}\sum_{\substack{\pi\in f_n\\ \chi_n^\pi=0}}\sum_{\mathcal{G}_n^\gamma\subset\mathcal{P}_n^\pi} G_{b_\gamma} D_{\to n}^\gamma \\ &+ \sum_{n\in\mathcal{F}_s}\sum_{\substack{\pi\in f_n\\ \chi_n^\pi=1}}\sum_{\mathcal{G}_n^\gamma\subset\mathcal{P}_n^\pi} G_{b_\gamma} D_{*n}^\gamma(q_n^{\pi(\gamma)}) \\ &+ \lambda\cdot\sum_{n\in\mathcal{F}_s}\sum_{\substack{\pi\in f_n\\ \chi_n^\pi=1}} |q_n^\pi| \end{aligned} \tag{6}$$

Direct minimization of (6) is difficult because of the inter-dependencies that exist between predicted precincts and their predictors as has been shown in Section II. For example, the decision to make a given precinct, $\mathcal{P}_j^\pi$, in frame f_j predicted ($\chi_j^\pi = 0$) depends on the quality of its predictors, $\mathcal{A}(\mathcal{P}_j^\pi)$, but the quality of these predictors depends to some extent on χ_j^π; using the precincts in $\mathcal{A}(\mathcal{P}_j^\pi)$ for predicting $\mathcal{P}_j^\pi$ increases their associated additional contribution weights which results in the assignment of more bytes (higher quality) to the precincts in $\mathcal{A}(\mathcal{P}_j^\pi)$ in the Lagrangian optimization.

We deal with this difficulty in a way similar to that we employed in [4] and [5]; we start by utilizing the additive distortion model of (3) in (6) to get

$$\begin{aligned} J_\lambda = &\sum_{n\in\mathcal{F}_s}\sum_{\substack{\pi\in f_n\\ \chi_n^\pi=0}}\sum_{\mathcal{G}_n^\gamma\subset\mathcal{P}_n^\pi} (1+\theta_n^\gamma)\cdot G_{b_\gamma} D_{\to n}^{\mathrm{M},\gamma} \\ &+ \sum_{n\in\mathcal{F}_s}\sum_{\substack{\pi\in f_n\\ \chi_n^\pi=1}}\sum_{\mathcal{G}_n^\gamma\subset\mathcal{P}_n^\pi} (1+\theta_n^\gamma)\cdot G_{b_\gamma} D_{*n}^\gamma(q_n^{\pi(\gamma)}) \\ &+ \lambda\cdot\sum_{n\in\mathcal{F}_s}\sum_{\substack{\pi\in f_n\\ \chi_n^\pi=1}} |q_n^\pi| \end{aligned} \tag{7}$$

Here, we have decomposed the distortion in each grid block into its original sources, a combination of quantization distortion and motion distortion. The reader is referred to Example 1 at the end of Section II for a typical decomposition example, and Examples 2 for example on calculating θ_n^γ.

Then, we employ an iterative approach that has two passes: the contribution weight pass, Ψ_w; and the optimization pass, Ψ_o. In Ψ_w, we visit all the frames within the WOF $\mathcal{F}_s$ in the acyclic ordering[3] of the converse dependency WADG[4], updating each additional contribution weight, θ_n^γ, in each frame we visit, so that (7) correctly represents (6) subject to $\{\chi_n^\pi\}_\pi$ and $\{q_n^\pi\}_\pi$ remaining constant; we update each θ_n^γ using (21), as will be derived in Subsection IV-B.

In Ψ_o, we visit all the frames within $\mathcal{F}_s$ following the acyclic ordering of the original dependency WADG this time. In this pass, we select the values of $\{\chi_n^\pi\}_\pi$ and $\{q_n^\pi\}_\pi$ that minimize the cost functional of (7), while $\{\theta_n^\gamma\}_\gamma$ are kept constant.

To determine χ_n^π and q_n^π for a given precinct, $\mathcal{P}_n^\pi$, we need to identify the contribution of that precinct to the cost functional J_λ of (7). This contribution is discussed in Example 2 of Section II; that is, the effective distortion of a grid block, $\mathcal{G}_n^\gamma$, is $(1+\theta_n^\gamma)\cdot D_{\to n}^\gamma$ when $\mathcal{G}_n^\gamma$ is predicted and $(1+\theta_n^\gamma)\cdot D_{*n}^\gamma$ when $\mathcal{G}_n^\gamma$ is directly decoded. Therefore, for a precinct, $\mathcal{P}_n^\pi$, we write

$$\hat{D}_{*n}^\pi(q_n^\pi) = \sum_{\mathcal{G}_n^\gamma\subset\mathcal{P}_n^\pi} (1+\theta_n^\gamma)\cdot G_{b_\gamma} D_{*n}^\gamma(q_n^{\pi(\gamma)}) \tag{8}$$

and

$$\hat{D}_{\to n}^\pi = \sum_{\mathcal{G}_n^\gamma\subset\mathcal{P}_n^\pi} (1+\theta_n^\gamma)\cdot G_{b_\gamma} D_{\to n}^\gamma \tag{9}$$

for the weighted (effective) precinct distortion associated with the de-quantized samples, $\mathcal{P}_{*n}^\pi(q_n^\pi)$, and the weighted precinct distortion associated with the predicted samples, $\mathcal{P}_{\to n}^\pi$, respectively. Then, the effective cost contribution of a precinct, $\mathcal{P}_n^\pi$, to the cost functional of (7) is

$$J_{n,\lambda}^\pi = \begin{cases} \hat{D}_{\to n}^\pi, & \chi_n^\pi = 0 \\ \hat{D}_{*n}^\pi(q_n^\pi) + \lambda\cdot|q_n^\pi|, & \chi_n^\pi = 1 \end{cases} \tag{10}$$

Thus, for each precinct we visit in Ψ_o, we first update $\hat{D}_{\to n}^\pi$ to its latest value, then we select the values of χ_n^π and q_n^π that yield the lowest precinct cost, $J_{n,\lambda}^\pi$. Using this method, multiple iterations of $\Psi_w\Psi_o$ might be needed to achieve the lowest possible cost functional, J_λ. This iterative process converges when a Ψ_o pass does not change any of the $\{\chi_n^\pi\}_\pi$.

We showed in [5] that this two-pass iterative approach converges in the absence of motion compensation, at least to a local minimum. The argument in [5] is also applicable when motion compensation is employed since, in both cases, Ψ_w is not part of the rate-distortion optimization and the decisions made during Ψ_o to minimize $J_{n,\lambda}^\pi$ are based on the correct $\hat{D}_{\to n}^\pi$ value at the time that precinct is visited; $D_{\to n}^\gamma$ depends on precincts that have already been optimized during this Ψ_o, and θ_n^γ depends on precincts in frames that are yet to be visited so that their χ_i^π values have not changed since the time θ_n^γ was computed. The interested reader is referred to [4] or [5] for more details.

Next, we give a graphical interpretation and a corresponding solution to (10). Figure 5 depicts a typical rate-distortion curve for a precinct, $\mathcal{P}_n^\pi$. It can be easily shown that this curve is convex, since each precinct layer is made up of convex-by-construction code-block contributions [24]. The distortion-length slope associated with quality layer q_n^π for this precinct is $\lambda_n^\pi(q_n^\pi) = (D_{*n}^\pi(q_n^\pi - 1) - D_{*n}^\pi(q_n^\pi))/(|q_n^\pi| - |q_n^\pi - 1|)$. The existence of predicted samples with distortion $D_{\to n}^\pi$ modifies the effective distortion-length convex hull whenever $D_{\to n}^\pi < D_{*n}^\pi(0)$ as shown in Figure 5. Thus, the distortion-length slopes associated with the first few layers change to $\lambda_{\to n}^\pi$.

In the above, we have ignored the effect of additional contribution weights for simplicity. In practice, we work with

[3]It is always possible to arrange the vertices of a WADG in what is called *acyclic ordering* [23], where each node is positioned after all of its reference nodes and before any of its dependent nodes.

[4]For every WADG, there is a converse WADG that is obtained by reversing all the arcs of the original WADG [23], as shown in Figure 4b.

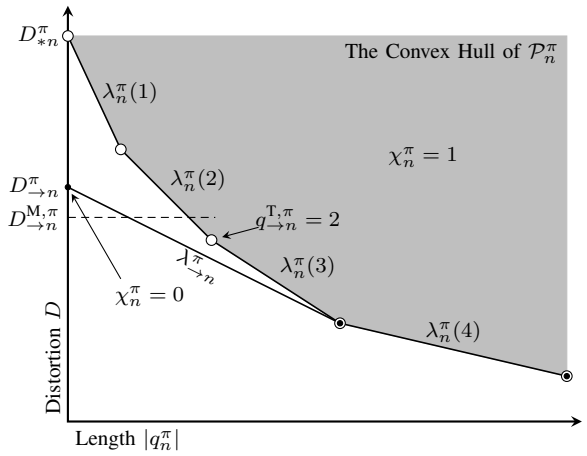

Fig. 5. A typical distortion-length convex hull for a precinct P^{π}_{n}, where each large white circle (○) represents one quality layer. Also shown in the figure is the distortion associated with the predicted version of the precinct, $D^{\pi}_{\rightarrow n}$, when $D^{\pi}_{\rightarrow n} < D^{\pi}_{*n}(0)$; the small black circles (•) represent the modified convex hull.

the terms $\hat{D}^{\pi}_{*n}(q^{\pi}_{n})$ and $\hat{D}^{\pi}_{\rightarrow n}$, as defined in (8) and (9), respectively, writing

$$\hat{\lambda}^{\pi}_{n}(q^{\pi}_{n}) = \frac{\hat{D}^{\pi}_{*n}(q^{\pi}_{n}-1) - \hat{D}^{\pi}_{*n}(q^{\pi}_{n})}{|q^{\pi}_{n}| - |q^{\pi}_{n}-1|} \tag{11}$$

for the weighted distortion-length slope associated with q^{π}_{n} quality layers and $\hat{\lambda}^{\pi}_{\rightarrow n}$ for the the weighted distortion-length slopes associated with the first few layers. Since $\hat{D}^{\pi}_{*n}(q^{\pi}_{n})$ depends upon multiple grid-block weights θ^{γ}_{n} (see (8)), it is possible that the $\hat{\lambda}^{\pi}_{n}(q^{\pi}_{n})$ terms are no longer monotonically decreasing, so the convexity of the precincts distortion-length characteristics is no longer guaranteed. In practice, however, this rarely occurs. Thus, the complete solution to the minimization of (10) for oracle policies becomes

$$\chi^{\pi}_{n} = \begin{cases} 1, & D^{\pi}_{\rightarrow n} > D^{\pi}_{*n}(0) \text{ or } \lambda \le \hat{\lambda}^{\pi}_{\rightarrow n} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

$$q^{\pi}_{n} = \begin{cases} \max\{q \,|\, \hat{\lambda}^{\pi}_{n}(q) > \lambda\}, & \chi^{\pi}_{n} = 1 \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

IV. Estimation of Distortion and Contribution Weights

At the server side, performing motion compensation and then directly calculating distortions is not practical due to the high computational requirements. It is important, therefore, to develop a suitable approach for approximating distortion. The problem of finding the weights, θ^{γ}_{n}, is closely related and must also be subjected to complexity limiting approximations. This section investigates these approximations.

Although the final result at the end of this derivation seems intuitive, it is very easy to make mistakes in the derivation. For this reason, we find it useful for the wider community to have access to this derivation. The derivation presented here is general in that it is done with expansive motion models in mind; only the final result is limited to the case of non-expansive translational models.

A. Distortion Propagation and Estimation

Figure 6 shows that a reference frame, f_r, is obtained by synthesizing its sub-band decomposition. The error in f_r can be expressed in terms of the errors at each location $\mathbf{k}$ in each of its sub-bands, b_r, as

$$\delta f_r = \sum_{b_r} \sum_{\mathbf{k}} \delta\mathcal{B}^{b_r}_{r}[\mathbf{k}] \cdot S^{b_r}_{\mathbf{k}}$$

where $S^{b_r}_{\mathbf{k}}$ denotes the relevant synthesis vectors (they are images). A predicted frame f_n is obtained from f_r by applying the motion mapping operator, $\mathcal{W}_{r\rightarrow n}$, at the highest available resolution, as shown in Figure 6. Since $\mathcal{W}_{r\rightarrow n}$ is a linear operator, the error contribution of f_r to the predicted frame f_n is

$$\delta f_{r\rightarrow n} = \sum_{b_r} \sum_{\mathbf{k}} \delta\mathcal{B}^{b_r}_{r}[\mathbf{k}] \cdot \mathcal{W}_{r\rightarrow n}(S^{b_r}_{\mathbf{k}})$$

The attributed error at location $\mathbf{p}$ (shown in Figure 6) in the predicted sub-band b_n of f_n, due to errors in location $\mathbf{k}$ of sub-band b_r can be obtained by applying the linear analysis operator $A^{b_n}_{\mathbf{p}}$ for sub-band b_n at location $\mathbf{p}$; that is,

$$\delta\mathcal{B}^{\mathrm{A},b_n}_{r\rightarrow n}[\mathbf{p}] = \sum_{b_r} \sum_{\mathbf{k}} \delta\mathcal{B}^{b_r}_{r}[\mathbf{k}] \cdot \left\langle \mathcal{W}_{r\rightarrow n}(S^{b_r}_{\mathbf{k}}), A^{b_n}_{\mathbf{p}} \right\rangle$$

Assuming that the attributed errors in the sub-bands are approximately uncorrelated[5], the distortion power for some region $\mathcal{R}_n$ around $\mathbf{p}$ in sub-band b_n, shown in gray in Figure 6, can then be approximated by

$$\begin{aligned} &\sum_{\mathbf{p}\in\mathcal{R}_n} \left|\delta\mathcal{B}^{\mathrm{A},b_n}_{r\rightarrow n}[\mathbf{p}]\right|^2 \\ &\quad\approx \sum_{b_r} \underbrace{\sum_{\mathbf{p}\in\mathcal{R}_n} \sum_{\mathbf{k}} \left|\delta\mathcal{B}^{b_r}_{r}[\mathbf{k}]\right|^2 \cdot \left\langle \mathcal{W}_{r\rightarrow n}(S^{b_r}_{\mathbf{k}}), A^{b_n}_{\mathbf{p}} \right\rangle^2}_{D^{b_r\rightarrow b_n}_{\mathcal{R}_n}} \end{aligned} \tag{14}$$

The fact that both the $\mathcal{W}_{r\rightarrow n}(S^{b_r}_{\mathbf{k}})$ and $A^{b_n}_{\mathbf{p}}$ operators have limited support with decaying envelopes means that $D^{b_r\rightarrow b_n}_{\mathcal{R}_n}$ depends mainly on the distortion contributions $\delta\mathcal{B}^{b_r}_{r}[\mathbf{k}]$ inside and around the region $\mathcal{R}_r$, being the projection of $\mathcal{R}_n$ onto sub-band b_r. Figure 6 shows that $\mathcal{R}_n$ has a projection in every sub-band in f_r. All of these projections are shown in light gray except for one, shown in dark gray; the darker projection is the focus of the next discussion, but that choice is arbitrary and, in fact, any projection can be used for this discussion. If $\mathcal{R}_r$ is small enough such that the distortion around it can be approximated by a uniform attributed noise power $D^{b_r}_{\mathcal{R}_r} / |\mathcal{R}_r|$, we have

$$D^{b_r\rightarrow b_n}_{\mathcal{R}_n} \approx \frac{D^{b_r}_{\mathcal{R}_r}}{|\mathcal{R}_r|} \cdot \sum_{\mathbf{p}\in\mathcal{R}_n} \underbrace{\sum_{\mathbf{k}} \left\langle \mathcal{W}_{r\rightarrow n}(S^{b_r}_{\mathbf{k}}), A^{b_n}_{\mathbf{p}} \right\rangle^2}_{G^{b_r\rightarrow b_n}_{\mathcal{W}(\mathbf{p})}} \tag{15}$$

Here, $G^{b_r\rightarrow b_n}_{\mathcal{W}(\mathbf{p})}$ represents a power gain which reflects the contribution of noise power around location $\mathbf{k} \approx \overleftarrow{\mathcal{W}}^{b_r\rightarrow b_n}_{r\rightarrow n}(\mathbf{p})$ in sub-band b_r (shown in Figure 6) to the attributed distortion

[5]This assumption was discussed in Section II.

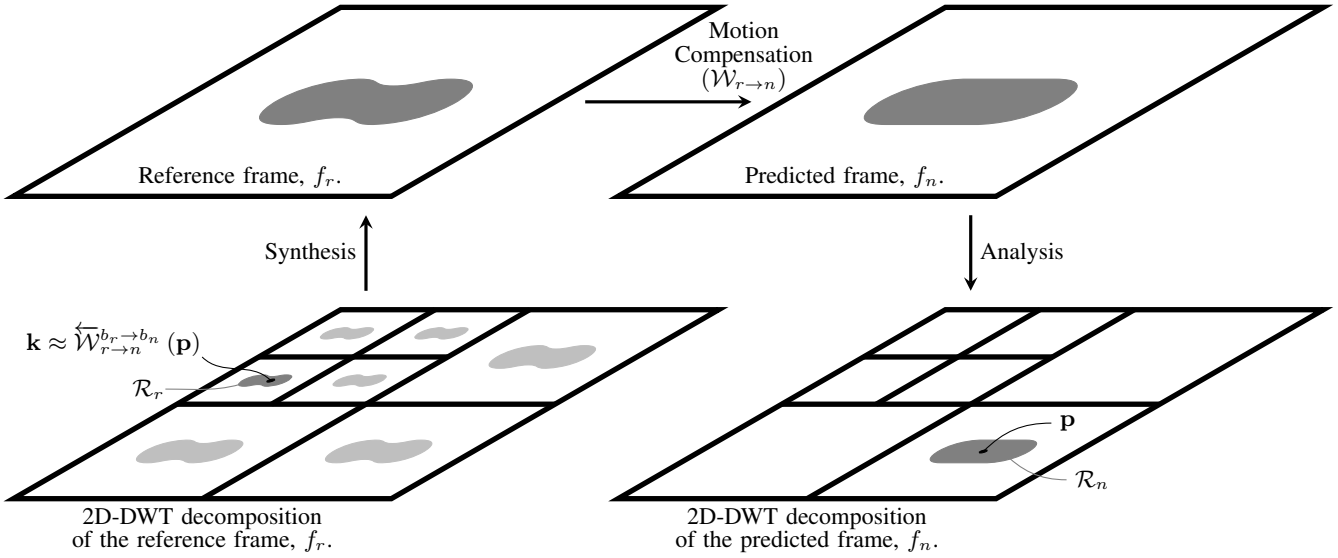

Fig. 6. The effect of distortion propagation from a reference frame f_r to a predicted frame f_n when the 2D-DWT is employed. In general, all the sub-bands in f_r contribute to the distortion in region $\mathcal{R}_n$ of frame f_n. Most of these contributions, however, come from the projections of region $\mathcal{R}_n$ onto the sub-bands of f_r, which are shown in gray in the 2D-DWT decomposition of f_r; here, we focus on one such region, $\mathcal{R}_r$. Region $\mathcal{R}_r$ encloses location $\overleftarrow{\mathcal{W}}_{r\to n}^{b_r\to b_n}(\mathbf{p})$ which corresponds to location $\mathbf{p}$ of $\mathcal{R}_n$.

at location $\mathbf{p}$ in sub-band b_n, where $\overleftarrow{\mathcal{W}}_{r\to n}^{b_r\to b_n}$ maps locations in sub-band b_n of frame f_n back to locations in sub-band b_r of the reference frame f_r, according to the motion model. Denoting the average noise power $D_{\mathcal{R}_r}^{b_r}/|\mathcal{R}_r|$ around $\mathcal{R}_r$ by $\bar{D}_r^{b_r}[\mathbf{k}]$ and the average attributed sub-band noise power around location $\mathbf{p}$ by $\bar{D}_{r\to n}^{\mathrm{A},b_n}[\mathbf{p}]$, (14) becomes

$$\bar{D}_{r\to n}^{\mathrm{A},b_n}[\mathbf{p}] \approx \sum_{b_r} \bar{D}_r^{b_r}\left[\overleftarrow{\mathcal{W}}_{r\to n}^{b_r\to b_n}(\mathbf{p})\right] \cdot \bar{G}_{\mathcal{W}(\mathbf{p})}^{b_r\to b_n} \qquad (16)$$

where $\bar{G}_{\mathcal{W}(\mathbf{p})}^{b_r\to b_n}$ is the average of the different values of $G_{\mathcal{W}(\mathbf{p})}^{b_r\to b_n}$ around point $\mathbf{p}$ because of the different phases[6] of $\mathbf{p}$. Thus, it is convenient to think of (16) as the *noise power propagation* from the area around the point that corresponds to $\mathbf{p}$ in the sub-bands of the reference frame, f_r, to the area around point $\mathbf{p}$ in the destination sub-band, with factors $\bar{G}_{\mathcal{W}(\mathbf{p})}^{b_r\to b_n}$ representing noise power gains.

Despite the fact that attributed noise can originate from any of the sub-bands in the reference frame, most of the attributed noise power in a given destination sub-band comes from source sub-bands that are at the same or similar decomposition levels in the reference frame [22]. Here, we formalize our selection of source sub-bands.

It can be seen from (15) that $\bar{G}_{\mathcal{W}(\mathbf{p})}^{b_r\to b_n}$ depends on the source sub-band, destination sub-band, motion compensation operator around $\mathbf{p}$, and the type of wavelet transform being used. The server is free to select amongst a variety of motion models (e.g. block-based translational model or mesh-based affine models); for these models, the server is free to choose coarse or fine block sizes. For prediction references, the server is also free to consider only one frame or employ some position-dependent linear combination of more than one nearby frame. Figure 7 depicts the case of a server that can, at serve time, choose a prediction model from a few possible models. For the convenience of this work, we focus only on the block-based translational model. Thus, for a given source sub-band, destination sub-band, and type of wavelet transform, the value of $\bar{G}_{\mathcal{W}(\mathbf{p})}^{b_r\to b_n}$ is a cyclo-stationary function of the spatial shift employed by the motion compensation operator. It is always possible to find a maximum value for $\bar{G}_{\mathcal{W}(\mathbf{p})}^{b_r\to b_n}$ over the set of possible spatial shifts, which we denote by $\bar{G}_{\max}^{b_r\to b_n}$.

For a given destination sub-band, the criterion we employ

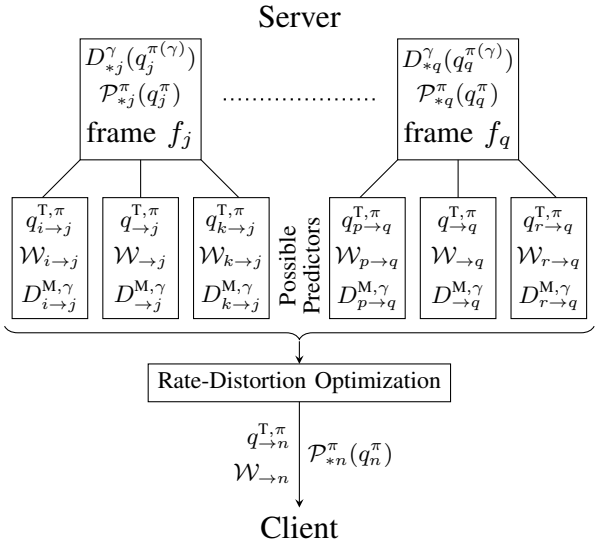

Fig. 7. The server can potentially explore more than one prediction model for a given frame and select the most appropriate one. To do that the server needs to store $D_{*n}^{\gamma}(q_n^{\pi(\gamma)})$ and $\mathcal{P}_{*n}^{\pi}(q_n^{\pi})$ for each frame, and $q_{\to n}^{\mathrm{T},\pi}$, $\mathcal{W}_{\to n}$, and $D_{\to n}^{\mathrm{M},\gamma}$ for each predictor. Only $\mathcal{P}_{*n}^{\pi}(q_n^{\pi})$, $q_{\to n}^{\mathrm{T},\pi}$, and $\mathcal{W}_{\to n}$ are delivered to the client.

[6]When sub-band b_r is from a coarser resolution (lower frequency) than sub-band b_n, the value of $G_{\mathcal{W}(\mathbf{p})}^{b_r\to b_n}$ changes slightly from one $\mathbf{p}$ point to the next depending on the phase of $\mathbf{p}$. Since we are only interested in an approximate distortion, averaging these values is sufficient.

is to only select the source sub-bands for which $\bar{G}_{\max}^{b_r \to b_n}$ is greater than or equal to a *Significance Threshold*, $T_{\mathcal{S}}$; that is, the set of source sub-bands that have b_n as their destination sub-band, denoted by $\mathcal{A}(b_n)$, is given by

$$\mathcal{A}(b_n) = \left\{ b_r | \bar{G}_{\max}^{b_r \to b_n} \geq T_{\mathcal{S}} \right\} \tag{17}$$

Table I shows the set of source sub-bands associated with each destination sub-band, for the cases $T_{\mathcal{S}} = 0.25$, $T_{\mathcal{S}} = 0.10$, and $T_{\mathcal{S}} = 0.05$, when a translational motion model is employed with CDF 9/7 irreversible wavelet transform and 5 levels of spatial decomposition. We discuss the impact of the significance threshold on the accuracy of distortion estimation in Section VIII.

For convenience of implementation, we approximate $\bar{D}_r^{b_r}[\mathbf{k}]$ as constant over grid blocks, $\mathcal{G}_r^{\gamma}$, writing $\bar{D}_r^{b_r}[\mathbf{k}] = D_r^{\gamma_r} / |\mathcal{G}_r^{\gamma_r}|$ for all $\mathbf{k} \in \mathcal{G}_r^{\gamma_r}$. We similarly approximate $\bar{D}_{r \to n}^{\mathrm{A},b_n}[\mathbf{p}]$, writing $\bar{D}_{r \to n}^{\mathrm{A},b_n}[\mathbf{p}] = D_{r \to n}^{\mathrm{A},\gamma_n} / |\mathcal{G}_n^{\gamma_n}|$ for the attributed distortion in grid block $\mathcal{G}_n^{\gamma_n}$ due to errors in frame f_r. Moreover, we use the motion model to directly map[7] index γ_n from sub-band b_n to index γ_r in sub-band b_r; that is, $\gamma_r = \overleftarrow{\mathcal{W}}_{r \to n}^{b_r \to b_n}(\gamma_n)$. Under these conditions, (16) can be recast as

$$\frac{D_{r \to n}^{\mathrm{A},\gamma_n}}{|\mathcal{G}_n^{\gamma_n}|} \approx \sum_{\substack{b_r \in \mathcal{A}(b_n) \\ \gamma_r = \overleftarrow{\mathcal{W}}_{r \to n}^{b_r \to b_n}(\gamma_n)}} \frac{D_r^{\gamma_r}}{|\mathcal{G}_r^{\gamma_r}|} \cdot \bar{G}_{\mathcal{W}(\gamma_n)}^{b_r \to b_n} \tag{18}$$

where $\bar{G}_{\mathcal{W}(\gamma_n)}^{b_r \to b_n}$ is $\bar{G}_{\mathcal{W}(\mathbf{p})}^{b_r \to b_n}$ at grid block $\mathcal{G}_n^{\gamma_n}$ that contains location $\mathbf{p}$. Since our grid blocks all have the same size, (18) becomes

$$D_{r \to n}^{\mathrm{A},\gamma_n} \approx \sum_{\substack{b_r \in \mathcal{A}(b_n) \\ \gamma_r = \overleftarrow{\mathcal{W}}_{r \to n}^{b_r \to b_n}(\gamma_n)}} D_r^{\gamma_r} \cdot \bar{G}_{\mathcal{W}(\gamma_n)}^{b_r \to b_n} \tag{19}$$

Having estimated $D_{r \to n}^{\mathrm{A},\gamma_n}$ using (19), these estimates are employed in (3) to find $D_{\to n}^{\gamma}$.

B. Estimation of Contribution Weights

The problem of estimating the additional contribution weights, θ_n^{γ}, is clearly related to the distortion estimation problem above.

We write $\mathcal{S}(b_r)$ for the set of sub-bands in a predicted frame, f_n, that use b_r as their prediction reference; that is,

$$\mathcal{S}(b_r) = \left\{ b_n | \bar{G}_{\max}^{b_r \to b_n} \geq T_{\mathcal{S}} \right\} \tag{20}$$

We write $\chi_n^{\pi(\gamma_n)}$ for the hidden state variable associated with grid block $\mathcal{G}_n^{\gamma_n}$ such that $\chi_n^{\pi(\gamma_n)} = \chi_n^{\pi}$ for all $\mathcal{G}_n^{\gamma_n} \subset \mathcal{P}_n^{\pi}$; in view of (19) and (20), and denoting the set of frames that are predicted from f_r by $\mathcal{S}(f_r)$, the additional contribution weight for grid block $\mathcal{G}_r^{\gamma_r}$ is

$$\theta_r^{\gamma_r} = \sum_{n \ni f_n \in \mathcal{S}(f_r)} g_{rn}^2 \cdot \sum_{\substack{b_n \in \mathcal{S}(b_r) \\ \gamma_n = \overrightarrow{\mathcal{W}}_{r \to n}^{b_r \to b_n}(\gamma_r) \\ \chi_n^{\pi(\gamma_n)} = 0}} \frac{G_{b_n}}{G_{b_r}} \cdot \bar{G}_{\mathcal{W}(\gamma_n)}^{b_r \to b_n} \cdot (1 + \theta_n^{\gamma_n}) \tag{21}$$

[7]Here, we are mapping power from a reference sub-band, b_r, to a destination sub-band, b_n; therefore, if more than one index γ_r maps to the same γ_n (for example, when b_r is from a finer resolution in the wavelet decomposition), it is sufficient to select one representative distortion from b_r; ideally, the γ_r index that maps to the center of the $\mathcal{G}_n^{\gamma_n}$.

where $\overrightarrow{\mathcal{W}}_{r \to n}^{b_r \to b_n}(\gamma_r)$ maps index γ_r in the reference sub-band, b_r, to index γ_n in predicted sub-band, b_n, using the motion model, and G_{b_n} and G_{b_r} are the energy gain factors of sub-bands b_n and b_r, respectively. Although the last equation looks complicated, its interpretation is simple. For each index γ_r in sub-band b_r of frame f_r, we find all the indices γ_n in sub-bands $\mathcal{S}(b_r)$ of all the frames $\mathcal{S}(f_r)$ that are predicted ($\chi_n^{\pi(\gamma_n)} = 0$) and we add their contribution, $g_{rn}^2 \cdot \frac{G_{b_n}}{G_{b_r}} \cdot \bar{G}_{\mathcal{W}(\gamma_n)}^{b_r \to b_n} \cdot (1 + \theta_n^{\gamma_n})$, to form $\theta_r^{\gamma_r}$. The reader can also refer to Section II for an example, Example 2, on evaluating θ_n^{γ}.

During Ψ_w, (21) is evaluated progressively by traversing the converse of the dependency WADG (see Figure 4b).

It is important to note that we use the same motion model for both $\overleftarrow{\mathcal{W}}_{r \to n}^{b_r \to b_n}$ and $\overrightarrow{\mathcal{W}}_{r \to n}^{b_r \to b_n}$; only the choice of the independent variable is different.

We discuss storage requirements and computational cost for distortion and contribution weight estimation in Section VI.

V. Actual Client and Server Policies and Side-Information Delivery

In this section, we discuss the actual client policy, actual server policy, and how side-information is delivered.

A. Actual Client Policy

The loose-coupling of client and server policies, first discussed in the introduction, requires any side-information that is sent to the client to be universal, by which we mean information that describes some properties of the video sequence being streamed that are always true and independent of the state of the client-server interaction. These properties should allow the client to make reasonably correct decisions with a wide diversity of contents, including those where the server is not fully aware of the client's cache contents.

Here, we propose a client policy and a corresponding server policy that are based on such a universal property, the per-precinct *quality layer threshold*, $q_{\to n}^{\mathrm{T},\pi}$. This threshold, shown in Figure 5, is the first quality layer at which it is better to use received samples than to use predicted samples assuming unquantized prediction source precincts. Specifically,

$$q_{\to n}^{\mathrm{T},\pi} = \min \left\{ q \mid D_{*n}^{\pi}(q_n^{\pi}) < D_{\to n}^{\mathrm{M},\pi} \right\} \tag{22}$$

We remind the reader that $D_{\to n}^{\mathrm{M},\pi}$ is obtained from full quality reference frames, and as such, $D_{\to n}^{\mathrm{M},\pi}$ represents the best possible result that prediction can be expected to produce using this prediction model. With this definition, the proposed client policy is

$$\mathcal{P}_n^{\pi} = \begin{cases} \mathcal{P}_{*n}^{\pi}(q_n^{\pi}), & q_n^{\pi} \geq q_{\to n}^{\mathrm{T},\pi} \\ \mathcal{P}_{\to n}^{\pi}, & \text{otherwise} \end{cases} \tag{23}$$

Obviously, the quality layer threshold is related to the motion compensated prediction model, and therefore each prediction model produces a different threshold. To keep things simple in this work, we choose to limit the possible prediction models for a given precinct to one. Thus, when one frame is predicted from two nearby frames, as in the case of hierarchical B-frames, the only possible predictor is the average of these two frames; this means that we only

TABLE I
EXAMPLE SHOWING SOURCE SUB-BANDS FOR EACH DESTINATION SUB-BAND FOR 3 DIFFERENT SIGNIFICANCE THRESHOLDS, T_S, WHEN TRANSLATIONAL MOTION MODEL IS EMPLOYED WITH CDF 9/7 IRREVERSIBLE WAVELET TRANSFORM AND 5 LEVELS OF SPATIAL DECOMPOSITION.

Source Sub-band		Destination Sub-band																	
		$d=5$	$d=4$			$d=3$			$d=2$			$d=1$			$d=0$		
		LL	HL	LH	HH	HL	LH	HH	HL	LH	HH	HL	LH	HH	HL	LH	HH
$d=5$	LL	⊗	⊙	⊙													
$d=4$	HL	⊗	⊗	○	⊙	⊗											
	LH	⊗	○	⊗	⊙		⊗										
	HH	⊙	⊗	⊗	⊗	⊗	⊗	○									
$d=3$	HL		⊙		⊙	⊗	○	⊙	⊗								
	LH			⊙	⊙	○	⊗	⊙		⊗							
	HH				⊙	⊗	⊗	⊗	⊗	⊗	○						
$d=2$	HL					⊙		⊙	⊗	○	⊙	⊗					
	LH						⊙	⊙	○	⊗	⊙		⊗				
	HH							⊙	⊗	⊗	⊗	⊗	⊗	○			
$d=1$	HL								⊙		⊙	⊗	○	⊙	⊗		
	LH									⊙	⊙	○	⊗	⊙		⊗	
	HH										⊙	⊗	⊗	⊗	⊗	⊗	○
$d=0$	HL											⊙		⊙	⊗	○	⊙
	LH												⊙	⊙	○	⊗	⊙
	HH													⊙	⊗	⊗	⊗

Note that ⊗, •, and ○ indicate T_S of 0.25, 0.10, and 0.05, respectively, and d is the decomposition level, where $d=5$ is the smallest resolution.

need one threshold for each precinct. In general, JSIV has the flexibility to employ a wide variety of prediction models, including position-dependent linear combinations of two or more frames, mesh-based affine prediction models, overlapped block-based prediction models, or even a combination of more than one model.

B. Actual Server Policy

Server optimization is done in epochs; each epoch corresponds to a fixed time step and a fixed amount of data to be transmitted. In each epoch, p, all the frames within the corresponding WOF have a chance of contributing data to the transmission. It is possible that one WOF is optimized over more than one consecutive epoch.

We write $q_n^{p,\pi}$ for the number of quality layers at the end of epoch p; we initialize $q_n^{0,\pi}$ to the number of quality layers in the client cache that the server is aware of. In order for the client to use the data it receives from the server for a given precinct, that data must achieve the requirements set out in the first case of (23); that is, $q_n^{p,\pi} \geq q_{\rightarrow n}^{\mathrm{T},\pi}$. This client policy changes the distortion-length slope associated with the first few quality layers whenever $q_{\rightarrow n}^{\mathrm{T},\pi} > 0$; in this case, the first point in the effective distortion-length characteristics for precinct $\mathcal{P}_n^\pi$ becomes $(0, \hat{D}_{\rightarrow n}^\pi)$ and the second point becomes $(|q_{\rightarrow n}^{\mathrm{T},\pi}|, \hat{D}_{*n}^\pi(q_{\rightarrow n}^{\mathrm{T},\pi}))$ which may not belong to the convex hull. If we denote the distortion-length slope that is associated with the first two points on the convex hull of the effective distortion-length characteristics by $\acute{\lambda}_{\rightarrow n}^\pi$, then the server's optimization process is driven by

$$\chi_n^\pi = \begin{cases} 1, & q_{\rightarrow n}^{\mathrm{T},\pi} = 0 \text{ or } \lambda \leq \acute{\lambda}_{\rightarrow n}^\pi \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

This way the server policy works with the client policy to attempt to achieve (4) by making it more favorable for the client to use lower distortion options.

C. Quality Layer Thresholds Delivery

In practice, it is not required for the client to receive all the quality layer thresholds, $q_{\rightarrow n}^{\mathrm{T},\pi}$, for all the precincts in each frame; especially when limited bandwidth is available. Therefore, we send these thresholds only for some of the precincts, as explained next.

Many ways exist to send the quality layer thresholds to the client, but we propose to send them as one additional JPEG2000 image component per prediction model inside each frame of the video sequence. This allows the use of JPIP without any modifications for sending this information to the client; it also allows us to benefit from features of JPEG2000 such as efficient compression, scalability, and progressive refinement in communicating this information.

Obviously, the quality layer thresholds component is heavily sub-sampled since there is only one threshold per precinct of the regular image components. We use the same number of decomposition levels and quality layers, Q, to compress the thresholds component. In fact, even the code-block dimensions used to compress the thresholds component are the same as those used for original frame data, although this is not necessary. Only one sub-band is needed to store all the thresholds for each resolution level; in practice, we use the HL band, leaving the LH and HH bands zero.

The thresholds are encoded using the JPEG2000 block encoder directly. We set the number of coding passes to $3 \cdot Q - 2$ and encode $q_{\rightarrow n}^{\mathrm{T},\pi}$ as $2^{Q - q_{\rightarrow n}^{\mathrm{T},\pi}}$. The resulting code-stream is constructed in such a way that each quality layer stores one whole bit-plane.

Side information is delivered to the client using the standard JPIP protocol. We send enough quality layers (or bit-planes)

from the thresholds component such that the client is able to deduce $q^{\mathrm{T},\pi}_{\rightarrow n}$ for all the precincts that have $q^{\pi}_{n} \geq q^{\mathrm{T},\pi}_{\rightarrow n}$; this makes it favorable for the client to use the received samples for these precincts. Thus, for a code-block, C, from the quality layer thresholds component, the number of layers, ℓ^{C}_{n}, transmitted is

$$\ell^{C}_{n} = 1 + \max_{\pi \in C} \left\{ q^{\mathrm{T},\pi}_{\rightarrow n} \mid q^{\pi}_{n} \geq q^{\mathrm{T},\pi}_{\rightarrow n} \right\} \tag{25}$$

VI. Storage Requirements and Computational Cost

Storage requirement and computational cost are related; therefore, it is more convenient to present them together.

A. Computational Cost

We consider, here, the computation cost at the server. We do not include the cost of motion estimation because it is not part of the server problem; i.e., we do not perform motion estimation for each client being served. Motion estimation is part of the pre-processing stage as shown in Figure 1.

Here, we denote the average number of elements in $\mathcal{A}(b_n)$ of (17) by $\mathcal{A}_{T_S}$, the average of the number of elements in $\mathcal{S}(b_n)$ of (20) by $\mathcal{S}_{T_S}$. Using approximate calculation in the rate-distortion optimization pass, Ψ_o, each predicted grid block distortion, $D^{\gamma}_{\rightarrow n}$, requires approximately $(\mathcal{A}_{T_S} + 1) \cdot |\mathcal{A}(f_n)|$ multiplications and $\mathcal{A}_{T_S} \cdot |\mathcal{A}(f_n)|$ additions for (19) and (3).

For the modified convex hull analysis, we need to find $\hat{D}^{\pi}_{*n}(q^{\pi}_{n})$ and $\hat{D}^{\pi}_{\rightarrow n}$, as defined in (8) and (9), respectively; $\hat{D}^{\pi}_{*n}(q^{\pi}_{n})$ requires $2 \cdot Q$ multiplications and $2 \cdot Q$ additions per grid block, where Q is the number of quality layers, while $\hat{D}^{\pi}_{\rightarrow n}$ requires 2 multiplications and 2 additions per grid block. For the modified convex hull analysis itself, the Incremental Computation of Convex Hull and Slopes algorithm presented in [24] requires no more than $2 \cdot Q$ multiplications per *precinct*. We could reduce the computational cost associated with finding $\hat{D}^{\pi}_{*n}(q^{\pi}_{n})$ by averaging the contribution weights, θ^{γ}_{n}, in precinct $\mathcal{P}^{\pi}_{n}$ and multiplying this average by precomputed unweighted precinct distortions, $D^{\pi}_{*n}(q^{\pi}_{n})$. Early termination strategies could also be employed to further reduce computational requirements.

For Ψ_w, each grid block contribution weight, θ^{γ}_{r}, requires approximately $2 \cdot \mathcal{S}_{T_S} \cdot |\mathcal{S}(f_n)|$ multiplications and $2 \cdot \mathcal{S}_{T_S} \cdot |\mathcal{S}(f_n)|$ additions for (21), assuming the $\frac{G_{b_n}}{G_{b_r}}$ terms are pre-calculated.

It can be seen that the computational cost for a given frame is inversely proportional to the grid block size as both of $|\mathcal{A}(f_n)|$ and $|\mathcal{S}(f_n)|$ are either 1 or 2 for the prediction models explored in Section VII, and both $\mathcal{A}_{T_S}$ and $\mathcal{S}_{T_S}$ are between 4 and 7 (see Table I). Experimental results reveal that grid blocks of 16×16 are sufficient; we discuss the effects of distortion approximation and grid block size on the quality of reconstructed video in more detail in Section VIII. Thus, for a grid block that represents 256 samples, a computational cost of a few tens of multiplications and additions is sufficient. This is significantly less than doing the actual motion compensation and then directly calculating distortions. Obviously, the computational cost here is higher than that of JSIV without motion compensation, as presented in [5], but it can be reduced to only a few times more that that of [5]. Importantly, computational cost grows linearly with the frame size.

B. Storage Requirements

To implement approximate distortion calculations the server needs to keep tables of grid block quantization distortions, $D^{\gamma}_{*n}(q^{\pi(\gamma)}_{n})$, for all quality layers, q^{π}_{n}, and grid block motion distortions, $D^{\mathrm{M},\gamma}_{\rightarrow n}$. The server also needs to keep a table of quality layer thresholds, $q^{\mathrm{T},\pi}_{\rightarrow n}$; there is no need to keep a table for quality layer lengths, $|q^{\pi}_{n}|$, as these can be easily obtained from code-block headers.

Representing distortions by 2 bytes is sufficient because the inaccuracy due to the additive model of (3) is usually larger than the inaccuracy in such a representation. Thus, $2Q + 2$ bytes are needed per grid block, and one byte per precinct for the quality layer threshold. The number of bytes needed per grid block can be significantly reduced with a simple compression algorithm since the higher frequency sub-bands do not usually make any contribution in the initial quality layers. More research is needed to find a more efficient way of storing this data, but that is beyond the scope of this work.

VII. Experimental Results

Three sequences[8] are used in this work, the standard "Crew" and "City" test sequences and the "Aspen" test sequence[9]. Both "Crew" and "City" have 193 frames[10] with a resolution of 704×576 and a bit depth of 8 bits per sample. The "Crew" sequence has a frame rate of 60 frames/s while "City" has 30 frames/s. "Aspen" is a 97 frame sequence[11] that has a resolution of 1920×1024[12] at 30 frames/s and a bit depth of 8 bits per sample. Only the Y-component is used for all the tests reported here.

For JSIV, the sequences are converted to JPEG2000 using Kakadu[13]. Five levels of irreversible DWT are employed for all the sequences. A code-block size of 32×32 and 20 quality layers are used for all sequences. "Hierarchical" refers to a 3-level hierarchical B-frame prediction arrangement, similar to the SVC extension of H.264 [3] and is denoted by JSIV-H. In the "Sequential" prediction arrangement (denoted by JSIV-S), each frame is predicted from the frame before it; effectively an "IPP…" arrangement. In JSIV-S, the server jointly optimizes two consecutive frames (WOF $= 2$) at each optimization epoch, and then shifts the WOF by one frame. For INTRA,

[8] "Crew," "City," and "Aspen" test sequences are available at http://www.eet.unsw.edu.au/~taubman/sequences.htm.

[9] "Aspen" test sequence is owned by NTIA/ITS, an agency of the U.S. Federal Government, and is available at ftp://vqeg.its.bldrdoc.gov/HDTV/NTIA_source/

[10] The original sequences are actually a little longer but only the first 193 are used. The length of 193 is selected because it is suitable for a 3-level hierarchical B-frame prediction arrangement.

[11] The original sequence has 600 frames but only the first 97 were used to reduce processing time. The length of 97 is selected because it is suitable for a 3-level hierarchical B-frame prediction arrangement.

[12] The original sequence has a resolution of 1920×1080, but was cropped due to limitations in the motion encoding implementation.

[13] http://www.kakadusoftware.com/, Kakadu software, version 5.2.4.

also known as Motion-JPEG2000, each frame is independently transmitted in an optimal fashion.

JSIV provides great flexibility in selecting motion models. So instead of using a single model, it is possible to work with a family of motion models, representing a variety of trade-offs between motion quality and bit-rate, selecting an appropriate model for each client. This work demonstrate this flexibility with a simple example; we employ an embedded scalable motion encoder [25], [26] to produce a block-based motion description that contains geometry information. The encoder employs Lagrange-style rate-distortion optimization. At the coarsest level the block size is 64×64 while at the finest level it is 8×8 for the hierarchical B-frames arrangement. For the sequential arrangement, block size ranges from 32×32 to 4×4. In each case, we have 4 possible motion descriptions with varying degrees of quality for each prediction arrangement. Motion compensation is performed at $1/4$ pixel precision with 7-tap interpolation kernels formed by windowing cubic splines. As mentioned before, motion compensation is always applied to synthesized frames at the highest available resolution.

Using an embedded scalable description of motion makes it possible to progressively refine the motion vectors with the availability of more bandwidth; for example, when the client browses the same media a second or a third time. Ideally, the motion vector quality should be related to the quality of media being served, but this feature is not yet implemented in our code. Therefore, we manually select one of the four possible motion descriptions such that motion information constitutes around 10% of the overall data rate wherever possible.

To find $q_{\rightarrow n}^{\mathrm{T},\pi}$ of (22), we employ the best quality motion vectors. This is fair since this choice matches the original definition of $q_{\rightarrow n}^{\mathrm{T},\pi}$ in that it is the first quality layer at which it is better to use received samples than to use predicted samples assuming unquantized prediction source precincts.

For SVC, JSVM[14] is used to compress and reconstruct the sequences. The intra-frame period is set to 8 to match that of JSIV. All the scenarios presented here employ three levels of temporal decimation with two enhancement layers. The enhancement layers use two levels of medium-grain scalability (MGS) between them, giving a total of seven quality layers. No spatial scalability options are used for these tests; these would penalize the SVC performance somewhat.

All results are reported in PSNR calculated from the average MSE over the reconstructed sequence. All JSIV results reported use the policies of Section V with 3 passes of $\Psi_w \Psi_o$ for the hierarchical B-frame prediction arrangement and 2 for the sequential. The results presented here are obtained using approximate distortion calculations with 4×4 grid block dimensions and a significance threshold, $T_{\mathcal{S}}$, of 0.05. The rates reported include all encoded sub-band samples, JPEG2000 headers, side information, motion information, and JPIP message header overhead. The only missing overhead is the one associated with motion information delivery; this is because we have not yet encapsulated motion information in a manner directly suitable for JPIP delivery.

We compare JSIV with SVC because it is considered to be the state of the art compressor with support for scalability. The results presented here are biased in favor of SVC, since they do not account for the communication overhead needed to stream SVC, e.g., using RTP. By contrast, JSIV results include all overhead associated with the highly flexible JPIP protocol.

We start by comparing JSIV performance against that of SVC and INTRA. Figures 8, 9, and 10 show the PSNR for the "Crew," "City," and "Aspen" sequences, respectively. It can be seen that both JSIV and SVC perform better than INTRA. SVC performs better than JSIV in the case of "Crew" and "City" while JSIV-H performs comparably or slightly better than SVC for the "Aspen" sequence. The good performance of JSIV-H for the "Aspen" sequence is due to the effectiveness of the motion compensation for that sequence and the fact that the "Apsen" sequence has large smooth regions (regions with very little high-frequency content). For such regions, JSIV sends nothing or very little from high-frequency sub-bands while SVC needs to send the many macro-blocks in these regions.

In general, JSIV performs better for high-resolution sequences because 32×32 code-blocks provide better accessibility at these resolutions; when a certain region of a predicted frame needs to be updated (such as when the motion model fails), a 32×32 code-block represents a small region in a high-resolution frame while it represents a very substantial portion of the frame for low-resolution sequences (the whole LL sub-band can be one code-block).

It is important to remember that JSIV is a relatively new concept whereas predictive video coding research has produced a lot of ideas in the last three decades that significantly improved the quality of reconstructed video. For example, JSIV currently uses fixed scaling factors g_{rn} of 0.5 in (1) to mix forward and backward prediction terms together in the hierarchical B-frame arrangement, as opposed to position-dependent scaling factors applied to reference frames in SVC.

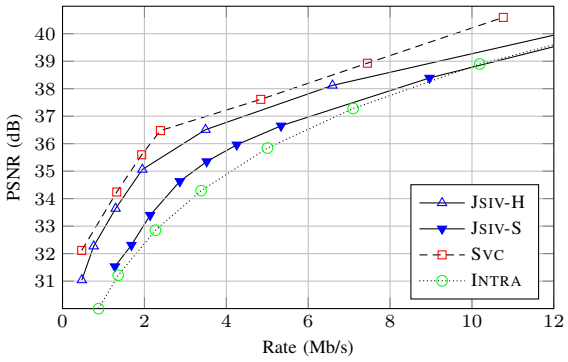

Fig. 8. A comparison of the performance of various schemes for the "Crew" sequence.

It can also be seen that the performance of JSIV-H is better than JSIV-S. This can be explained by the fact that the hierarchical arrangement produces better predictors compared to sequential.

Other than motion information, the overheads associated with these test sequences are usually less than 10%; motion

[14]JSVM version 9.19.7 obtained through CVS from its repository at garcon.ient.rwth-aachen.de

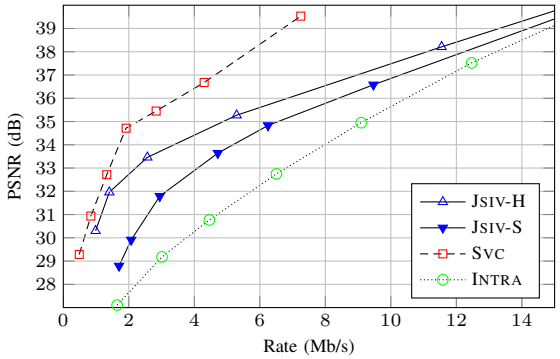

Fig. 9. A comparison of the performance of various schemes for the "City" sequence.

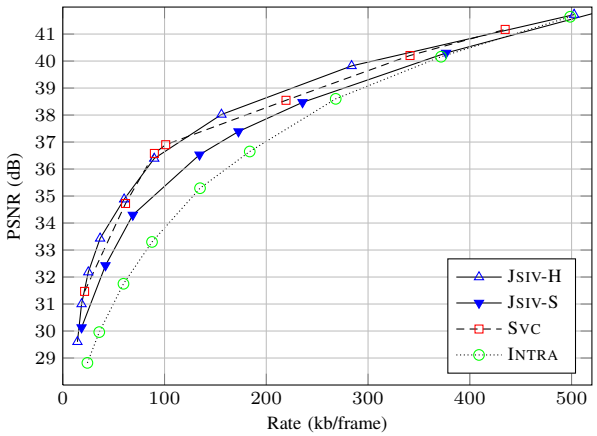

Fig. 10. A comparison of the performance of various schemes for the "Aspen" sequence. Note that the x-axis is in (kb/frame).

information, on the other hand, can be a significant portion of the overall data rate, mainly because the available four motion descriptions are not sufficient to cover the wide range of data rates we are employing (for example, from 1 Mb/s to 20Mb/s). Table II shows the overheads associated with the sequential prediction arrangement of the "City" sequence, measured against the overall rate, where side-information refers to the quality layer thresholds. Table III shows the overheads associated with the hierarchical B-frame arrangement of the "Aspen" sequence.

TABLE II
OVERHEADS IN JSIV FOR THE "CITY" SEQUENCE IN SEQUENTIAL ARRANGEMENT AS A PERCENTAGE OF THE OVERALL RATE.

Rate (Mb/s)	JPIP	Side Information	JPIP for Side Information	Motion Information
1.151	0.348%	4.404%	0.270%	12.041%
2.073	0.431%	2.941%	0.182%	9.726%
4.718	0.866%	1.489%	0.084%	5.053%
6.251	1.035%	1.228%	0.051%	3.814%
9.469	1.023%	0.867%	0.040%	2.518%
19.738	0.706%	0.508%	0.021%	1.051%

Examining the overheads reveals that the percentage of motion information decreases with the increase in data rate; this is partially due to the increase in data information, but, more importantly, is also due to the increased dependence on

TABLE III
OVERHEADS IN JSIV FOR THE "ASPEN" SEQUENCE IN HIERARCHICAL B-FRAMES ARRANGEMENT AS A PERCENTAGE OF THE OVERALL RATE.

Rate (kb/frame)	JPIP	Side Information	JPIP for Side Information	Motion Information
18.446	2.091%	7.932%	0.663%	23.597%
41.973	1.851%	4.638%	0.379%	10.370%
68.814	1.850%	3.528%	0.230%	7.859%
172.658	1.565%	1.792%	0.091%	3.132%
376.730	1.361%	0.880%	0.039%	1.406%
749.935	1.172%	0.454%	0.020%	0.022%

directly decoded precincts at higher rates. As more precincts become directly decoded, as opposed to predicted, motion information becomes irrelevant and can be safely discarded. The results also suggest that more research is needed to produce an embedded motion model that can support a wide range of data rates; such a model can perhaps improve JSIV results.

Next, we consider the effect of using different code-block sizes on the quality of reconstructed video. Figure 11 and Figure 12 show the PSNR for the "Crew" and "City" sequences, respectively, when the hierarchical B-frame arrangement is used. It can be seen that code-block dimensions of 32×32 provide the best compromise between accessibility and coding efficiency. This result is, to a large extent, similar to the result obtained for the case of JSIV without motion compensation [5].

In our earlier work [5], we demonstrated the efficacy of JSIV without motion compensation under several usage scenarios. These included: individual frame retrieval; spatial and temporal scalability; window of interest; and the use of client-cached data in improving received data. All of these scenarios can also be employed in JSIV with motion compensation; however, we choose not to repeat the same experiments here. Instead, we choose two new scenarios to demonstrate the flexibility of JSIV.

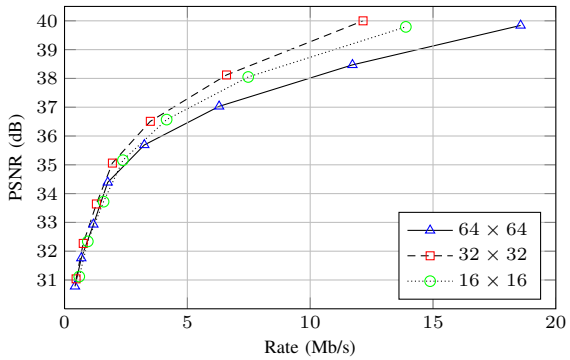

Fig. 11. The effect of code-block size on the quality of reconstructed video for the "Crew" sequence with hierarchical B-frame arrangement.

The first scenario involves a client that already has a better motion model than the model currently being delivered by the server, possibly from an earlier browsing session; these models are from the same embedded motion model that is mentioned earlier. For this case, the client can use its better motion model to obtain a higher quality reconstructed video;

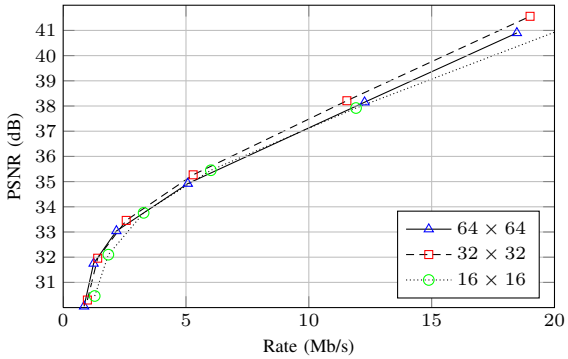

Fig. 12. The effect of code-block size on the quality of reconstructed video for the "City" sequence with hierarchical B-frame arrangement.

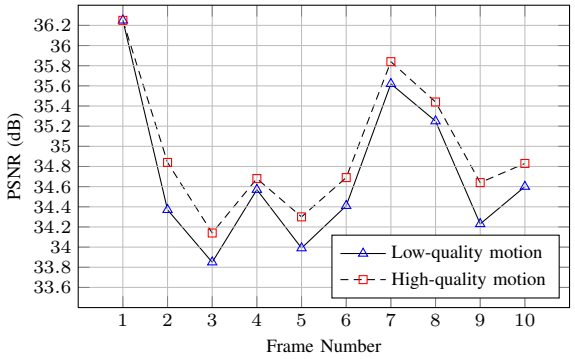

Fig. 13. A demonstration of the flexibility of JSIV. A client can immediately utilize the availability of better motion vectors in improving the quality of reconstructed video. The figure shows the PSNR for the first 10 frames of the "Aspen" sequence; "Sequential" prediction arrangement is used here.

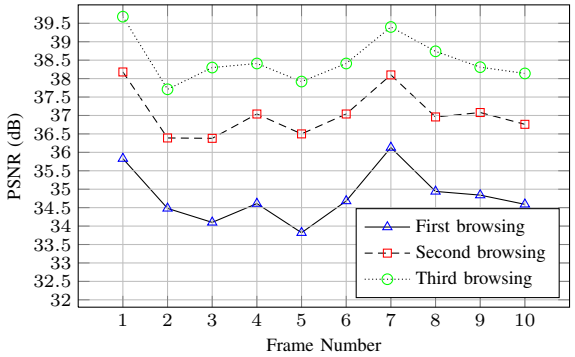

Fig. 14. A demonstration of the flexibility of JSIV. A client that browses the same section of a video will progressively get improved quality. The figure shows the PSNR for the first 10 frames of the "Aspen" sequence. "Sequential" prediction arrangement is used here. Each frame receives around 10.5 kBytes at each browsing session.

the client has the finest available description with blocks of 4×4 while the server is delivering the coarsest available description with blocks of 32×32. Figure 13 shows the PSNR of reconstructed video with these two different motion model qualities, but for the same encoded sub-band samples from the first 10 frames of the "Aspen" sequence, using the sequential prediction arrangement. It can be seen that the availability of a better motion description improves the quality of reconstructed video. It is important to note that this is not possible with traditional predictive coding because side information is tightly-coupled to the motion residues.

We have demonstrated in [5] that it is possible for a client to benefit from receiving new data even if the server is not aware of the client's cache contents. Such a scenario is also applicable when motion compensation is employed; however, we choose not to repeat it here. The scenario we present here shows how a server that is aware of the client's cache contents can use this knowledge to improve reconstructed video quality when the client revisits the same part of the video a second and third time by augmenting the client's cache contents. Figure 14 shows the PSNR of reconstructed video after the first, second, and third visit to the first 10 frames of the "Aspen" sequence with a sequential prediction arrangement, starting from the first frame each time. The data rate allocated for each frame in each pass is around 10.5 kBytes. It can be seen that the availability of higher quality sub-band samples greatly improves the quality of reconstructed video. This scenario is not currently available with traditional predictive coding techniques but an SVC server can be modified to operate in such a scenario.

VIII. Impact of Distortion Approximations on the Quality of Reconstructed Video

In order to achieve a realistic implementation of JSIV, we have introduced a few approximations; in this section, we investigate the effects of these approximations on the quality of reconstructed video. We also study the effect of using our actual client and server policies instead of oracle policies.

We introduced approximate distortion estimation (referred to here as Approx) in Section IV to reduce the computational cost associated with exact distortion calculations (referred to here as Exact). Experimental results from Tables IV and V for the headings Exact and Approx show that the degradation in reconstructed video quality is more for the sequential prediction arrangement than that for the hierarchical arrangement; this is true at low bit rates for both test sequences investigated here, where video reconstruction is more dependent on prediction compared to higher bit rates. We attribute this degradation to the accumulation of errors that happens when there are multiple consecutive predictions (a frame is predicted from a frame that is itself predicted and so on); multiple predictions occur in the sequential arrangement more than in the hierarchical which can at most have 3 consecutive predictions. This impact becomes smaller as the data rate increases since the client and server policies become less dependent on prediction with the increase in data rate; thus, the impact of distortion approximations becomes more acceptable at the practical PSNR region[15] with a maximum loss of around 0.6 dB.

Comparing these results to the case of JSIV without motion compensation presented in [5], we see that the inaccurate assumption of uncorrelated motion distortion has a lower impact when motion compensation is used because motion compensation tends to make motion distortion smaller; more-

[15]A video sequence with a PSNR of less than 33dB is considered of very poor quality.

TABLE IV
A COMPARISON BETWEEN DIFFERENT POLICIES FOR THE "CITY" SEQUENCE.

Rate[a]	INTRA	Sequential				Hierarchical B-frames			
		Oracle Policy		Actual Policy		Oracle Policy		Actual Policy	
(Mb/s)		EXACT	APPROX	EXACT	APPROX	EXACT	APPROX	EXACT	APPROX
1	26.24	29.15	26.91	27.31	26.78	30.53	30.54	30.57	30.58
2	28.22	30.95	29.63	30.67	29.85	32.99	33.01	32.99	33.01
4	30.94	33.04	32.38	32.99	32.52	34.71	34.69	34.71	34.69
8	34.81	35.89	35.61	35.88	35.67	36.89	36.84	36.89	36.84
12	37.86	38.10	37.98	38.10	37.99	38.83	38.80	38.83	38.80
16	40.46	40.52	40.51	40.52	40.51	40.66	40.64	40.66	40.64

[a] To provide a fair comparison, all results reported here are for encoded sub-band samples and motion information only; they exclude any headers, JPIP, and policy overhead.

TABLE V
A COMPARISON BETWEEN DIFFERENT POLICIES FOR THE "ASPEN" SEQUENCE.

Rate[a]	INTRA	Sequential				Hierarchical B-frames			
		Oracle Policy		Actual Policy		Oracle Policy		Actual Policy	
(kb/frame)		EXACT	APPROX	EXACT	APPROX	EXACT	APPROX	EXACT	APPROX
20	28.55	31.23	30.73	30.75	30.63	31.69	31.67	31.69	31.68
40	30.74	33.18	32.59	32.85	32.68	34.12	34.02	34.11	34.05
80	33.44	35.58	34.96	35.46	35.26	36.14	36.01	36.12	36.06
160	36.53	37.80	37.34	37.78	37.55	38.53	38.38	38.51	38.42
320	39.87	40.25	40.02	40.23	40.04	40.59	40.47	40.58	40.49
480	41.82	41.85	41.82	41.85	41.82	41.91	41.85	41.90	41.86
640	43.02	43.00	42.99	43.00	42.99	42.97	42.94	42.97	42.95

[a] To provide a fair comparison, all results reported here are for encoded sub-band samples and motion information only; they exclude any headers, JPIP, and policy overhead.

over, it mixes inaccurate motion distortion estimates with the more reliable quantization distortion values. Consider for example grid block $\mathcal{G}_k^1$ of Figure 4a; its distortion estimate combines reliable quantization distortions (from $\mathcal{G}_j^4$) with a less reliable combination of distortions (from $\mathcal{G}_j^2$).

Two parameters affect the approximation quality: the grid block size and the significance threshold. Smaller grid blocks and significance thresholds produce more accurate distortion estimates but increase the computational cost. The grid block size has a large impact on the computational cost, and therefore it is a good idea to maximize it. Experimental results for the sequential prediction arrangement of the "City" sequence show that a grid block size of 32×32 reduces the quality of reconstructed video by up to 0.5 dB while a size of 16×16 incurs a loss of at most 0.1 dB. The impact of grid block size is smaller for the hierarchical prediction arrangement of the "City" sequence and for both prediction arrangements of the "Aspen" test sequence. Based on these observations, we recommend a grid block size of 16×16.

The significance threshold factor has a rather low impact on the computational cost, but it is still a good idea to maximize it. For the sequential arrangement, experimental results reveal that increasing $T_{\mathcal{S}}$ from 0.05 to 0.1 has little effect on the quality of reconstructed video; however, increasing $T_{\mathcal{S}}$ from 0.1 to 0.25 can reduce the quality of reconstructed video by up to 1.5 dB at low bit-rates while having little effect at high bit-rates. For the hierarchical prediction arrangement, this effect is smaller. Based on these results, we recommend keeping $T_{\mathcal{S}}$ at or below 0.1.

Finally, we explore the effect of using actual client and server policies instead of oracle ones. Experimental results, shown in Tables IV and V under the "EXACT" heading, reveal that the PSNR difference between the oracle and actual policies is small in the practical PSNR region; this is true for both prediction arrangements of both test sequences, "City" and "Aspen". We conclude that our proposed client and server policies provide at least close to the best performance that can be practically achieved, noting that the oracle policies represent an unachievable upper bound on performance.

IX. CONCLUSION AND FUTURE WORK

In this work, we have demonstrated the efficacy of JSIV when motion compensation is employed. In general, the use of motion compensation improves prediction whenever the actual underlying motion can be modeled reasonably well; otherwise, JSIV effectively reverts back to intra-coded video. A hierarchical B-frame arrangement provides better exploitation of temporal redundancy compared to sequential arrangement; however, the same content can be simultaneously served to clients with different prediction strategies (e.g. to satisfy delay constraints). The computational cost of distortion estimation can be made reasonable through the use of appropriate approximations, allowing the server to perform rate-distortion optimization in real-time. Storing side information as meta-images allows the use of the standard JPIP protocol to send this information as well as streaming the video itself.

JSIV, with or without motion compensation, provides considerably better interactivity compared to existing streaming

schemes. This improved interactivity comes from not committing to a predetermined prediction policy. This allows the server to dynamically and adaptively change its policy to track a client's needs. The use of loosely-coupled policies makes it possible for the client and the server to work independently, which is especially beneficial for cases where the server cannot immediately be aware of the client's cache contents. Performance-wise, JSIV is comparable or slightly inferior to existing schemes in certain scenarios while performing better in those which are interactive in nature, such as video conferencing and remote browsing of videos.

This work is part of an ongoing investigation in this area. Future work includes improved prediction and a real-time prototype implementation of the JSIV client and server applications.

ACKNOWLEDGMENT

The authors would like to thank Dr. Reji Mathew for providing the source code for his embedded scalable motion encoder [25], [26] used for helping with the generation of motion vectors for this work.

REFERENCES

[1] J.-R. Ohm, "Advances in scalable video coding," *Proc. of the IEEE*, vol. 93, January 2005.

[2] N. Mehrseresht and D. Taubman, "An efficient content-adaptive motion compensated 3D-DWT with enhanced spatial and temporal scalability," *IEEE Trans. Image Proc.*, pp. 1397–1412, June 2006.

[3] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, September 2007.

[4] A. T. Naman, "JPEG2000-based scalable interactive video (JSIV)," Ph.D. dissertation, School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW 2052, Australia, accepted with minor correction.

[5] A. T. Naman and D. Taubman, "JPEG2000-based scalable interactive video (JSIV)," *Image Processing, IEEE Transactions on*, Accepted (available in the early access section).

[6] ——, "Optimized scalable video transmission based on conditional replenishment of JPEG2000 code-blocks with motion compensation," *MV '07: Proceedings of the International Workshop on Workshop on Mobile Video*, pp. 43–48, Septmeber 2007.

[7] ISO/IEC 15444-9, "Information technology – JPEG 2000 image coding system – Part 9: Interactivity tools, APIs and protocols," 2004.

[8] D. Taubman and R. Prandolini, "Architecture, philosophy and performance of jpip: internet protocol standard for JPEG 2000," *Int. Symp. Visual Comm. and Image Proc.*, vol. 5150, pp. 649–663, July 2003.

[9] ISO/IEC 15444-3, "Information technology – JPEG 2000 image coding system – Part 3: Motion JPEG 2000," 2007.

[10] A. T. Naman and D. Taubman, "A novel paradigm for optimized scalable video transmission based on JPEG2000 with motion," *Proc. IEEE Int. Conf. Image Proc. 2007*, pp. V–93 – V–96, Septmeber 2007.

[11] ——, "Rate-distortion optimized delivery of JPEG2000 compressed video with hierarchical motion side information," *Proc. IEEE Int. Conf. Image Proc. 2008*, pp. 2312–2315, October 2008.

[12] ——, "Distortion estimation for optimized delivery of JPEG2000 compressed video with motion," *IEEE 10th Workshop on Multimedia Signal Processing, 2008, MMSP 2008*, pp. 433–438, October 2008.

[13] ——, "Rate-distortion optimized JPEG2000-based scalable interactive video (JSIV) with motion and quantization bin side-information," *Proc. IEEE Int. Conf. Image Proc. 2009*, pp. 3081–3084, November 2009.

[14] ——, "Predictor selection using quantization intervals in JPEG2000-based scalable interactive video (JSIV)," *Proc. IEEE Int. Conf. Image Proc. 2010*, September 2010, accepted for publication.

[15] N.-M. Cheung and A. Ortega, "Flexible video decoding: A distributed source coding approach," *IEEE 9th Workshop on Multimedia Signal Processing, 2007, MMSP 2007.*, pp. 103–106, October 2007.

[16] ——, "Compression algorithms for flexible video decoding," *Visual Communications and Image Processing 2008*, vol. 6822, no. 1, p. 68221S, 2008.

[17] P. Zanuttigh, N. Brusco, D. Taubman, and G. Cortelazzo, "A novel framework for the interactive transmission of 3D scenes," *Signal Processing: Image Communication, Special Issue on Interactive Representation of Still and Dynamic Scenes*, vol. 21, no. 9, pp. 787 – 811, 2006.

[18] F.-O. Devaux, J. Meessen, C. Parisot, J.-F. Delaigle, B. Macq, and C. De Vleeschouwer, "A flexible video transmission system based on JPEG2000 conditional replenishment with multiple references," *Proc. IEEE Int. Conf. Acoust. Speech and Sig. Proc.*, April 2007.

[19] ——, "Remote interactive browsing of video surveillance content based on JPEG 2000," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 19, no. 8, pp. 1143–1157, August 2009.

[20] A. Mavlankar, J. Noh, P. Baccichet, and B. Girod, "Peer-to-peer multicast live video streaming with interactive virtual pan/tilt/zoom functionality," *Proc. IEEE Int. Conf. Image Proc. 2008*, pp. 2296–2299, October 2008.

[21] Y. Andreopoulos, A. Munteanu, J. Barbarien, M. V. der Schaar, J. Cornelis, and P. Schelkens, "In-band motion compensated temporal filtering," *Signal Processing: Image Communication*, vol. 19, no. 7, pp. 653 – 673, 2004, special Issue on Subband/Wavelet Interframe Video Coding. [Online]. Available: http://www.sciencedirect.com/science/article/B6V08-4CJV714-1/2/bfe0043c326ca402d49b1ae68ac91656

[22] N. Mehrseresht and D. Taubman, "A flexible structure for fully scalable motion compensated 3D-DWT with emphasis on the impact of spatial scalability," *IEEE Trans. Image Proc.*, pp. 740–753, Mar 2006.

[23] J. Bang-Jensen and G. Gutin, *Digraphs: Theory, Algorithms and Applications*, 2nd ed., ser. Springer monographs in mathematics. London: Springer-Verlag, 2009.

[24] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Boston: Kluwer Academic Publishers, 2002.

[25] R. Mathew and D. S. Taubman, "Quad-tree motion modeling with leaf merging," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 10, pp. 1331–1345, October 2010.

[26] R. Mathew, "Quad-tree motion models for scalable video coding applications," Ph.D. dissertation, School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW 2052, Australia, 2009. [Online]. Available: http://handle.unsw.edu.au/1959.4/44600

Aous Thabit Naman received B.Sc. degree in Electronics and Telecommunication Engineering from Al-Nahrain University, Baghdad, Iraq, in 1994, M.Eng.Sc. degree in Engineering from University of Malaya, Kuala Lumpur, Malaysia, in 2000. He is currently studying for the Ph.D. degree in Electrical Engineering at the School of Electrical Engineering and Telecommunications, Faculty of Engineering, the University of New South Wales, Sydney, Australia.

David Taubman (M'92) received the B.S. and B.Eng. degrees from the University of Sydney, Sydney, Australia, in 1986 and 1988, respectively, and the M.S. and Ph.D. degrees from the University of California, Berkeley, in 1992 and 1994, respectively.

From 1994 to 1998, he was with Hewlett-Packard's Research Laboratories, Palo Alto, CA, joining the University of New South Wales, Sydney, in 1998, where he is a Professor with the School of Electrical Engineering and Telecommunications. He is the coauthor, with M. Marcellin, of the book *JPEG2000: Image Compression Fundamentals, Standards and Practice* (Boston, MA: Kluwer, 2001). His research interests include highly scalable image and video compression, inverse problems in imaging, perceptual modeling, joint source/channel coding, and multimedia distribution systems.

Dr. Taubman was awarded the University Medal from the University of Sydney; the Institute of Engineers, Australia, Prize; and the Texas Instruments Prize for Digital Signal Processing, all in 1998. He has received two Best Paper awards, one from the IEEE Circuits and Systems Society for the 1996 paper, "A Common Framework for Rate and Distortion Based Scaling of Highly Scalable Compressed Video," and from the IEEE Signal Processing Society for the 2000 paper, "High Performance Scalable Image Compression with EBCOT."