Bit Allocation and Encoded View Selection for Optimal Multiview Image Representation

Gene Cheung [#], Vladan Velisavljević [o]

[#] *National Institute of Informatics*
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan 101-8430
[#] cheung@nii.ac.jp

[o] *Deutsche Telekom Laboratories*
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
[o] vladan.velisavljevic@telekom.de

Abstract—**Novel coding tools have been proposed recently to encode texture and depth maps of multiview images, exploiting inter-view correlations, for depth-image-based rendering (DIBR). However, the important associated bit allocation problem for DIBR remains open: for chosen view coding and synthesis tools, how to allocate bits among texture and depth maps across encoded views, so that the fidelity of a set of V views reconstructed at the decoder is maximized, for a fixed bitrate budget? In this paper, we present an optimization strategy to select subset of texture and depth maps of the original V views for encoding at appropriate quantization levels, so that at the decoder, the combined quality of decoded views (using encoded texture maps) and synthesized views (using encoded texture and depth maps of neighboring views) is maximized. We show that using the monotonicity property, complexity of our strategy can be greatly reduced. Experiments show that using our strategy, one can achieve up to 0.83dB gain in PSNR improvement over a heuristic scheme of encoding only texture maps of all V views at constant quantization levels. Further, computation can be reduced by up to 66% over a full parameter search approach.**

I. Introduction

In a typical multiple view imaging scenario, an image sequence of V views is captured by a set of closely spaced cameras. Because inter-view spatial correlations exist inherently, novel coding tools and structures [1], [2] for encoding of texture maps have been proposed to exploit this redundancy using disparity compensation for a compact representation of the captured V views.

Besides texture maps, depth information of a captured view (physical distance between camera and object corresponding to each captured pixel) can also be captured or estimated. Using pixel and depth maps of neighboring views, intermediate views can be synthesized via depth-image-based rendering (DIBR) [3] at high fidelity. Efficient coding tools for depth maps, with unique characteristics such as smooth surfaces and sharp edges, have also been proposed recently [4], [5].

Given efficient coding tools for texture and depth maps at the encoder and a view synthesis tool at the decoder, a natural question is the following: how to select texture and depth maps of a designated set of V captured views for encoding at encoder at appropriate quantization levels, so that at decoder, the distortion of the reconstructed V views—one subset are *decoded views* (decoded using corresponding encoded texture maps) and the other subset are *synthesized views* (synthesized using encoded texture and depth maps of neighboring views)—is minimized, subject to a rate constraint?

One can see that depending on the efficiency of the chosen coding and view synthesis tools and complexity of the captured scene, different optimal selections are possible. For example, if synthesis tool constructs views poorly and/or the captured scene is too complex for view interpolation, then encoding only texture maps for all V views (and no depth maps) is a good selection. On the other hand, if synthesized views can be constructed at sufficiently high fidelity, then encoding texture and depth maps for the first and last views only (1 and V)—at fine quantization levels for high-quality synthesized intermediate views—is a good selection. An optimal strategy should find the best selection possible for given desired rate-distortion (RD) tradeoff.

In this paper, we propose an optimization algorithm that finds the best possible subset of texture and depth maps of V captured views for encoding, and assigns appropriate quantization levels for selected maps. We first establish that the optimal selection of texture and depth maps for encoding at appropriate quantization levels is equivalent to the shortest path in a specially constructed three-dimensional (3D) trellis. Given that the state space of the 3D trellis is nonetheless enormous, we then show that using lemmas derived from monotonicity property in predictor's quantization level and distance, sub-optimal states and edges in the trellis can be eliminated respectively from consideration during shortest path calculation without loss of optimality. Experimental results show that optimal selection of texture and depth maps and associated quantization levels for encoding outperformed a heuristic scheme that selects only texture maps of all V views for coding and assigns fixed constant quantization levels for all maps by up to 0.83dB. Further, our algorithm can reduce computation complexity over full trellis calculation by up to

MMSP'10, October 4-6, 2010, Saint-Malo, France.
978-1-4244-8112-5/10/\$26.00 ©2010 IEEE.

66% without loss of optimality.

The paper is organized as follows. After describing related work in Section II, we formulate our bit allocation & view selection problem in Section III. Then, we discuss the construction of a corresponding 3D trellis, where an end-to-end shortest path corresponds to the optimal solution, and the important monotonicity property in Section IV. Using the discussed monotonicity property, we propose an efficient optimization algorithm in Section V. We present our experimental results in Section VI and conclude in Section VII.

II. Related Work

Novel tools for encoding texture maps [1], [2] and depth maps [4], [5] of multiview images have been recently proposed, but how bits should be optimally allocated among texture and depth maps for maximum fidelity is not addressed.

In [6], a view synthesis distortion model has been constructed and two quantization parameters have been have been assigned correspondingly, one for texture maps and one for depth maps. In contrast, our proposed scheme selects a unique quantization level for each chosen encoded map, taking dependent quantization into consideration, where a coarsely quantized predictor would lead to worse prediction, resulting in higher distortion and/or rate for the predicted view. Moreover, we do not construct any *analytical* models, but rely instead on real data (rate and distortion) collected using actual coding and view synthesis tools as the optimization is run. While our *operational* approach avoids modeling errors, the task of data collection can be overwhelming. Hence our focus is on complexity reduction so that only a minimal data set is required to find the optimal solution.

Optimal bit allocation among independent [7] and dependent [8] quantizers in an operational approach has been studied for RD-optimized media compression. Our work differs in that bit allocation for both pixel and depth maps are considered simultaneously, such that the resulting distortion of both encoded and synthesized views at the decoder is minimized for a desired RD tradeoff.

III. Formulation

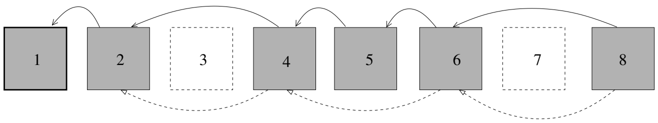


Fig. 1. An Example Selection of Coded (Gray) and Uncoded (White) Views. $\mathcal{J} = \{1, 2, 4, 5, 6, 8\}$, $\mathcal{J}^s = \{2, 4, 6, 8\}$, $\mathcal{J}' = \{3, 7\}$. Solid and dash arrows show texture and depth map dependencies, respectively.

The setting of our bit allocation problem is as follows. A desired set of views $\mathcal{V} = \{1, \ldots, V\}$ in a 1D-camera-array arrangement are to be conveyed from sender to receiver, using a set of chosen compression and view synthesis tools, at highest possible fidelity for a given bitrate constraint. Views $\mathcal{V}$ are to be optimally divided into K *coded views*, $\mathcal{J} = \{j_1, \ldots, j_K\}$, and $V - K$ *uncoded views* $\mathcal{J}' = \mathcal{V} \setminus \mathcal{J}$. The first and last view in $\mathcal{V}$ must be selected as coded views; i.e., $1, V \in \mathcal{J} \subseteq \mathcal{V}$. Texture and possibly depth maps of a coded view j_i are encoded using quantization level q_{j_i} and p_{j_i}, respectively. q_{j_i} and p_{j_i} take on discrete values from quantization level set $\mathcal{Q} = \{1, \ldots, Q_{\max}\}$ and $\mathcal{P} = \{0, 1, \ldots, P_{\max}\}$, respectively. We assume the convention that a larger q_{j_i} or p_{j_i} implies a coarser quantization, with the exception that $p_{j_i} = 0$ means no depth map is encoded for view j_i.

An uncoded view $j' \in \mathcal{J}'$ is not encoded at sender but is synthesized at receiver, using texture and depth maps of the closest left and right coded views, $l, r \in \mathcal{J}$. We will assume both texture and depth maps from the same closest coded views l and r are needed to synthesize uncoded view j' in-between.

We assume inter-view differential coding is performed separately for texture and depth maps. Texture map of view 1 is always encoded as I-frame. Texture maps of each subsequent coded view j_i—view 2, 4, 5, 6 and 8 in Fig. 1—are encoded as P-frame, each using texture map of previous coded view j_{i-1} as predictor for disparity compensation. Depth maps of coded views that are chosen for encoding are similarly differentially encoded. Note, however, that because not all depth maps of coded views are selected for encoding—coded view 1 and 5 in Fig. 1 are not used for view synthesis and hence their depth maps are not encoded—view dependency for depth maps is in generally different from view dependency for texture maps.

More formally, we define the subset of indices $\mathcal{J}^s$ where depth maps are chosen for encoding from the coded view indices $\mathcal{J}$ as follows:

$$\mathcal{J}^s = \{j \in \mathcal{J} \ \mid p_{j_i} > 0\} \tag{1}$$

Hence depth map of the first view j^s_1 in $\mathcal{J}^s$ will be coded as an I-frame, and subsequent depth maps $j^s_i > j^s_1$ will be coded as P-frames.

A. Visual Distortion

Given the coded view dependencies, we can now write distortion D^c of the coded views as a function of the texture map quantization levels, $\mathbf{q} = [q_{j_1} \ldots, q_{j_K}]$:

$$D^c(\mathbf{q}) = d^c_1(q_1) + \sum_{i=2}^{K} d^c_{j_i, j_{i-1}}(q_{j_i}, q_{j_{i-1}}) \tag{2}$$

(2) states that distortion d^c_1 of the starting I-frame depends only on its own texture quantization level q_1, while distortion $d^c_{j_i, j_{i-1}}$ of P-frame j_i depends on both its own texture quantization level q_{j_i} and its predictor j_{i-1}'s level $q_{j_{i-1}}$. A more general model [8] is to have P-frame j_i depends on its own q_{j_i} and all previous quantization levels $q_1, \ldots, q_{j_{i-1}}$. We assume here that truncating the dependencies to $q_{j_{i-1}}$ only is a good first-order approximation.

Similarly, we now write the distortion of the synthesized (uncoded) views D^s as a function of $\mathbf{q}$ and depth quantization levels, $\mathbf{p} = [p_{j_1}, \ldots, p_{j_K}]$:

$$\begin{aligned} D^s(\mathbf{q}, \mathbf{p}) &= \sum_{j' \in \mathcal{J}'} d^s_{j', l, r}(q_l, p_l, q_r, p_r) && (3) \\ l &= \arg\min_{j^s \in \mathcal{J}^s} |j' - j^s| && \text{s.t. } j^s < j' \\ r &= \arg\min_{j^s \in \mathcal{J}^s} |j' - j^s| && \text{s.t. } j^s > j' \end{aligned}$$

where l and r are indices of the closest coded views to the left and right of synthesized view j' with both texture and depth maps encoded. In words, distortion $d^s_{j',l,r}$ of synthesized view j' depends on both texture and depth map quantization levels of the two spatially closest coded views l and r, where $l, r \in \mathcal{J}^s$.

B. Encoding Rate

As done for distortion, we can write the rate of texture and depth maps of coded views, R^c and R^s, respectively, as follows:

$$R^c(\mathbf{q}) = r^c_1(q_1) + \sum_{i=2}^{K} r^c_{j_i,j_{i-1}}(q_{j_i}, q_{j_{i-1}}) \tag{4}$$

$$R^s(\mathbf{p}) = r^s_{j^s_1}(p_{j^s_1}) + \sum_{\forall j^s_i \in \mathcal{J}^s | j^s_i > j^s_1} r^s_{j^s_i,j^s_{i-1}}(p_{j^s_i}, p_{j^s_{i-1}}) \tag{5}$$

(4) states that the encoding rate $r^c_{j_i,j_{i-1}}$ for texture map of a coded view j_i depends on its texture map quantization level, q_{j_i}, and its predictor's level, $q_{j_{i-1}}$. Similarly, (5) states that the encoding rate $r^s_{j_i,j_{i-1}}$ for depth map of coded view j_i depends on its depth map quantization level, $p_{j^s_i}$, and its predictor's depth map level, $p_{j^s_{i-1}}$.

C. Rate-distortion Optimization

Given the above formulation, the optimization we are interested in is to find the coded view indices $\mathcal{J} \subseteq \mathcal{V}$, and associated texture and depth quantization vector, $\mathbf{q}$ and $\mathbf{p}$, such that the Lagrangian objective is minimized for given Lagrangian multiplier $\lambda \geq 0$:

$$\min_{\mathcal{J},\mathbf{q},\mathbf{p}} \Phi_\lambda = D^c(\mathbf{q}) + D^s(\mathbf{q},\mathbf{p}) + \lambda \left[R^c(\mathbf{q}) + R^s(\mathbf{p}) \right] \tag{6}$$

IV. Trellis and Monotonicity

We first show that the optimal solution to (6) can be computed by first constructing a *three-dimensional trellis* (3D trellis), and then finding the shortest path from the left end of the trellis to the right end using the famed Viterbi Algorithm (VA). Nevertheless, the complexity of constructing the full trellis is large; we will then discuss the important monotonicity property, using which a fast algorithm will be designed.

A. Full Trellis & Viterbi Algorithm

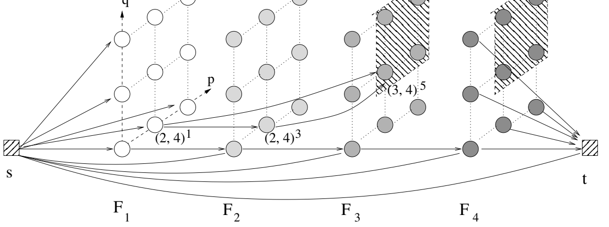


Fig. 2. 3D trellis for the selection of coded views with both texture and depth maps encoded. Start state s, end state t, and planes of states for four views are shown.

1) Trellis Construction: We can construct a 3D trellis—a four-view example is shown in Fig. 2—for the selection of coded views $\mathcal{J}^s$ with both texture and depth maps encoded, and corresponding texture and depth quantization levels $\mathbf{q}$ and $\mathbf{p}$, as follows. Each view $j \in \mathcal{V}$ is represented by a *plane* of states, where each state represents a pair of levels $(q_j, p_j)^j$ for texture and depth maps. Because each state $(q_j, p_j)^j$ indicates view j has both texture and depth map encoded, $p_j > 0$. In addition, there is a single start state s and an end state t, from which a path in the 3D trellis must start and end.

From each state $(q_j, p_j)^j$ of view j, there are forward edges to all states $(q_k, p_k)^k$ of view k, $k > j$. Selecting such an edge in an end-to-end path in the 3D trellis would mean view j and k are both selected as coded views with both texture and depth maps encoded, *and* views i's in-between, $j < i < k$, are *either* coded views with texture maps encoded only, *or* uncoded views to be synthesized using encoded texture and depth maps of view j and k at receiver. The exact configuration for each edge—which in-between views i's are selected as coded views and which are uncoded views—and the Lagrangian cost of the edge will be discussed in Section IV-A2.

There are also forward edges from start state s to all other states (including t), as well as forward edges from all states to end state t. Selecting an edge from s to a state $(q_j, p_j)^j$ of view j means views prior to view j are coded views with texture maps encoded only. Selecting an edge from a state $(q_k, p_k)^k$ of view k to t means views after view k are coded views with texture maps encoded only. We discuss edge cost in the 3D trellis next.

2) Calculating 3D Edge Cost: To calculate the Lagrangian cost of an edge in the 3D trellis from state $(q_j, p_j)^j$ of view j to $(q_k, p_k)^k$ of view k, $k > j$, we need to select in-between views i's, $j < i < k$, to be coded views with texture maps encoded only (at appropriate texture map quantization levels q's), with remaining views to be synthesized at receiver using encoded texture and depth maps of view j and k, such that the Lagrangian cost of this 3D edge is minimized. We accomplish this by constructing a corresponding *two-dimensional trellis* (2D trellis) and finding the shortest end-to-end path within it.

The 2D trellis is constructed as follows. Each in-between view i, $j < i < k$, is represented by a *column* of states $(q_i)^i$'s, one state $(q_i)^i$ for each texture map quantization level $q_i \in \mathcal{Q}$. Trellis has a start state $(q_j, p_j)^j$ for view j and an end state $(q_k, p_k)^k$ for view k. Each state of view i has forward edges to all states of view i', $i' > i$. An example is shown in Fig. 3.

To calculate edge costs for the 2D trellis, we first define $\phi_{j_i,j_{i-1}}(q_{j_i}, q_{j_{i-1}})$ to be the Lagrangian cost of coded view j_i using view j_{i-1} as predictor for differential texture map coding, i.e.,

$$\phi_{j_i,j_{i-1}}(q_{j_i}, q_{j_{i-1}}) = d^c_{j_i,j_{i-1}}(q_{j_i}, q_{j_{i-1}}) + \lambda r^c_{j_i,j_{i-1}}(q_{j_i}, q_{j_{i-1}}) \tag{7}$$

An edge from state $(q_i)^i$ of view i to state $(q_{i+1})^{i+1}$ of neighboring view $i+1$ will carry cost $\phi_{i+1,i}(q_{i+1}, q_i)$. Selecting such an edge in an end-to-end path in the 2D trellis would mean view $i+1$ is selected as coded view with texture

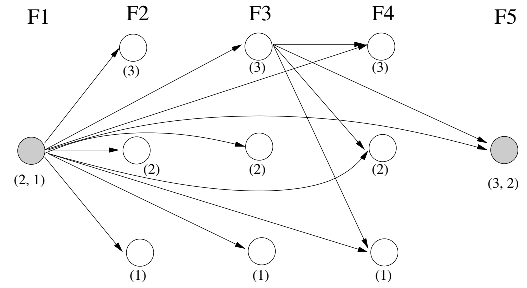


Fig. 3. Calculating a 3D edge cost via a 2D trellis. Only edges from start state $(2,1)^1$ and state $(3)^3$ are shown.

map quantization level q_{i+1}. An edge from state $(q_i)^i$ of view i to state $(q_{i'})^{i'}$ of a further-away view i' will carry similar cost $\phi_{i',i}(q_{i'}, q_i)$, *plus* synthesized view distortions $\sum_{i<x<i'} d^s_{x,j,k}(q_j, p_j, q_k, p_k)$. Selecting such an edge would mean view i' is selected as coded view with texture map level $q_{i'}$, *and* views between i and i' are uncoded and must be synthesized using texture and depth maps of view j and k.

The cost of the shortest path in the corresponding 2D trellis from start state $(q_j, p_j)^j$ of view j to end state $(q_k, p_k)^k$ of view k (found using VA also) *plus* the cost of encoding depth map of view k, $\lambda r^s_{k,j}(p_k, p_j)$, will be assigned the cost of the 3D edge in the original 3D trellis. Note that the shortest path in the corresponding 2D trellis means the best possible combination of coded and uncoded views are selected for in-between views i's, $j < i < k$, and each selected coded view i is assigned the best possible quantization level q_i in a Lagrangian sense.

3) Shortest Path in 3D Trellis: Having discussed the calculation of a 3D edge cost in the 3D trellis in Section IV-A2, the definition of the 3D trellis is complete. We argue that the shortest path in the 3D trellis, found using VA, is equivalent to the optimal solution to (6). The reason is quite straightforward; every possible selection (including the null set) of coded views with both texture and depth maps encoded, $\mathcal{J}^s$, with associated quantization level pairs $(q_j, p_j)^j$'s, can be represented by a series of edges in the 3D trellis. For given $\mathcal{J}^s$, every possible set of coded views with texture maps encoded only, with texture map quantization levels q's, for remaining views $\mathcal{V} \setminus \mathcal{J}^s$, are represented by a path through a series of 2D trellises corresponding to 3D edges connecting $\mathcal{J}^s$. Since the shortest path in the 3D trellis considers all possible selections of $\mathcal{J}^s$ and then all possible selections of $\mathcal{J} \setminus \mathcal{J}^s$, the optimal solution will be optimal to (6) as well.

Nevertheless, the number of states and edges in the 3D trellis is large: $O(|\mathcal{Q}||\mathcal{P}|V)$ and $O(|\mathcal{Q}|^2|\mathcal{P}|^2V^2)$, respectively. Hence the crux to reduce complexity is to find the shortest path by visiting only a small subset of states and edges. We first discuss the monotonicity property, using which a fast algorithm can be designed to simplify the shortest path search.

B. Monotonicity

Previous work [8] has shown that using monotonicity property of dependent quantizers, efficient algorithms and heuristics can be constructed for optimal or near-optimal bit allocation. Our work can be viewed as a generalization of [8] to include synthesized views. We first discuss the useful monotonicity property along different dimensions. We then derive lemmas based on monotonicity and construct a fast optimization algorithm using the lemmas in the next section.

1) Monotonicity in Predictor's Quantization Level: Motivated by a similar empirical observation in [8], we assume here also the *monotonicity in predictor's quantization level* for Lagrangian $\phi_{j_i,j_{i-1}}$ of coded view j_i and synthesized distortion $d^s_{j',l,r}$ of synthesized view j'; i.e., for any $\lambda \geq 0$:

$$\begin{array}{rcll}
\phi_{j_i,j_{i-1}}(q_{j_i}, q_{j_{i-1}}) & \leq & \phi_{j_i,j_{i-1}}(q_{j_i}, q^+_{j_{i-1}}) & (8) \\
d^s_{j',l,r}(q_l, p_l, q_r, p_r) & \leq & d^s_{j',l,r}(q^+_l, p_l, q_r, p_r) & (9) \\
d^s_{j',l,r}(q_l, p_l, q_r, p_r) & \leq & d^s_{j',l,r}(q_l, p^+_l, q_r, p_r) &
\end{array}$$

where q^+_n (or p^+_n) implies a larger (coarser) quantization level than q_n (or p_n). In words, (8) states that if predictor view j_{i-1} uses a coarser quantization level in texture map, it will lead to a worse prediction for view j_i, resulting in larger distortion and/or coding rate, and hence a larger Lagrangian cost $\phi_{j_i,j_{i-1}}$ for all values of $\lambda \geq 0$.

(9) makes a similar statement for monotonicity of the synthesized view distortion $d^s_{j'_i,l,r}$ with respect to the texture and depth map quantization levels q_l and p_l of the closest left coded view l. We assume also monotonicity in the texture and depth quantization levels q_r and p_r of the closest right coded view r as well.

2) Monotonicity in Predictor's Distance: We can also express monotonicity with respect to the *predictor's distance* for a coded view performing differential coding, or for an uncoded view performing view synthesis. Assuming further-away predictor view k^- for coded view j, $k^- < k$, has the same quantization level q_k as view k, and further-away predictor views l^- and r^+ have the same levels for synthesized view j' as respective levels of views l and r, we can write:

$$\begin{array}{rcll}
\phi_{j,k}(q_j, q_k) & \leq & \phi_{j,k^-}(q_j, q_k) & (10) \\
d^s_{j',l,r}(q_l, p_l, q_r, p_r) & \leq & d^s_{j',l,r^+}(q_l, p_l, q_r, p_r) & (11) \\
d^s_{j',l,r}(q_l, p_l, q_r, p_r) & \leq & d^s_{j',l^-,r}(q_l, p_l, q_r, p_r) &
\end{array}$$

Here, $r^+ > r$ or $l^- < l$ implies a further-right coded view r^+ or further-left coded view l^- is used to synthesize view j'. In words, (10) and (11) say that using a further-away predictor to differentially encode or synthesize a view, given the quantization levels of texture and depth maps of the further-away predictor are the same, results in no smaller Lagrangian cost or synthesized distortion. These inequalities hold true assuming Lambertian scenes.

V. Bit Allocation Optimization

A. Reducing Complexity in 2D Trellis

We first derive two lemmas based on the monotonicity property discussed earlier. Using the derived lemmas, we construct a computation-efficient algorithm to search for the shortest path in a 2D trellis corresponding to a 3D edge cost.

Suppose we are given a 2D trellis with start state $(q_j, p_j)^j$ of view j and end state $(q_k, p_k)^k$ of view k, $j < k$. Let $\Phi_i(q_i)$

1) Compute edge cost $(q_j, p_j)^j \rightarrow (q_i)^i$ for all states $(q_i)^i$'s and store as $\Phi_i(q_i)$. Compute edge cost $(q_j, p_j)^j \rightarrow (q_k, p_k)^k$ and store as $\Phi_k(q_k)$. Initialize $i = j + 1$.
2) Find q_i^* s.t. $\Phi_i(q_i^+) > \Phi_i(q_i^*)$, $\forall q_i^+ > q_i^*$. Eliminate states $(q_i^+)^i$'s from consideration.
3) For each survived state $(q_i)^i$ of view i, evaluate forward sub-paths to states $(q_{i+1})^{i+1}$'s of neighboring view $i + 1$.
4) For each survived state $(q_i)^i$ of view i, if $\phi_{i+1,i}(q_i, q_i) > d_{i+1,j,k}(q_j, p_j, q_k, p_k)$, then evaluate forward edges $(q_i)^i \rightarrow (q_{i'})^{i'}$ to states $(q_{i'})^{i'}$'s, $i' > i + 1$.
5) If $i < k - 1$, increment i by 1 and repeat step 2 to 4.

Fig. 4. Efficient Shortest Path Search for 2D Trellis corresponding to 3D Edge $(q_j, p_j)^j \rightarrow (q_k, p_k)^k$, $j < k$, in 3D Trellis.

be the cost of the shortest *sub-path* from start state $(q_j, p_j)^j$ of view j to state $(q_i)^i$ of view i in the 2D trellis, $j < i < k$. The first lemma eliminates *sub-optimal states* from consideration in the search for shortest path in the 2D trellis.

Lemma 1: Given view i, if $\Phi_i(q^+) > \Phi_i(q^*), \forall q^+ > q^*$, then states $(q^+)^i$'s, $\forall q^+ > q^*$, cannot belong to shortest path.

Proof of Lemma 1: We prove by contradiction. Suppose shortest path include state $(q^+)^i$, $q^+ > q^*$. We now reroute path via state $(q^*)^i$ instead of $(q^+)^i$ for view i. First, cost of sub-path to state $(q^*)^i$ is smaller than sub-path to $(q^+)^i$ by assumption. Further, Lagrangian cost of a coded view i' that used view i with level q^+ as predictor will be no worse using level $q^* < q^+$ instead by monotonicity in predictor's quantization level (8). Synthesized views (if any) use texture and depth maps of view j and k and hence are not affected by the reroute. Hence rerouting shortest path via state $(q^*)^i$ instead of $(q^+)^i$ yields strictly smaller cost. A contradiction. □

The second lemma eliminates *sub-optimal edges* from consideration in the search for shortest path in the 2D trellis.

Lemma 2: If Lagrangian cost of coding view $i + 1$, $\phi_{i+1,i}(q_i, q_i)$, at same quantization level as view i and using view i as predictor, is smaller than distortion of synthesizing view $i + 1$, $d_{i+1,j,k}(q_j, p_j, q_k, p_k)$, then edge from state $(q)^i$ to any state in view $i' > i + 1$ is sub-optimal.

Proof of Lemma 2: We prove by contradiction. Suppose a shortest path include an edge $(q_i)^i \rightarrow (q_{i'})^{i'}$. $i' > i + 1$. We now replace it with two edges $(q_i)^i \rightarrow (q_i)^{i+1} \rightarrow (q_{i'})^{i'}$. First, the Lagrangian cost of encoded view $i+1$ at quantization level q_i is smaller than distortion of synthesizing view $i + 1$ by assumption. Further, Lagrangian cost of coded view i' that used view i with level q_i as predictor will be no larger using a closer view $i + 1$ with the same level by monotonicity in predictor's distance (10). Synthesized views (if any) between view $i+1$ and i' use texture and depth maps of view j and k and hence are not affected by the edge replacement. Hence replacing edge $(q_i)^i \rightarrow (q_{i'})^{i'}$ with two edges $(q_i)^i \rightarrow (q_i)^{i+1} \rightarrow (q_{i'})^{i'}$ yields a strictly smaller cost. A contradiction. □

1) Efficient Algorithm for 2D Trellis: Using the two derived lemmas 1 and 2, we design a computation-efficient shortest path search algorithm, showed in Fig. 4, for a 2D trellis. It works as follows. First, we initialize cost of each shortest sub-path $\Phi_i(q_i)$ from start state $(q_j, p_j)^j$ to a state $(q_i)^i$ of view i as edge cost $(q_j, p_j)^j \rightarrow (q_i)^i$. Then, for each view i, we eliminate states $(q_i^+)^i$'s, where $\Phi_i(q_i^+) > \Phi_i(q_i^*)$ and $q_i^+ > q_i^*$, due to Lemma 1. We next "evaluate" edges $(q_i)^i \rightarrow (q_{i+1})^{i+1}$ to neighboring view $i + 1$ for survived states $(q_i)^i$'s. By evaluate, we mean comparing the cost of $\Phi_i(q_i)$ plus the edge cost $(q_i)^i \rightarrow (q_{i+1})^{i+1}$ to current cost $\Phi_{i+1}(q_{i+1})$, and updating $\Phi_{i+1}(q_{i+1})$ to the minimum of the two. We then evaluate edges from i to further-away views i', $i' > i + 1$, only if $\phi_{i+1,i}(q_i, q_i) > d_{i+1,j,k}(q_j, p_j, q_k, p_k)$, due to Lemma 2. The procedure repeats until end of trellis.

B. Reducing Complexity in 3D Trellis

We now derive two similar lemmas based on the monotonicity property to reduce search complexity in the 3D trellis. Let $\Phi_j(q_j, p_j)$ be the shortest sub-path from start state s to state $(q_j, p_j)^j$ of view j. The first lemma eliminates *sub-optimal states* $(q_j, p_j)^j$'s, given computed $\Phi_j(q_j, p_j)$'s, using monotonicity in quantization level.

Lemma 3: If at state plane of view j_i, for given p_{j_i}, $\Phi_{j_i}(q_{j_i}^+, p_{j_i}) > \Phi_{j_i}(q_{j_i}^*, p_{j_i})$, $\forall q_{j_i}^+ > q_{j_i}^*$, then sub-paths up to states $(q_{j_i}^+, p_{j_i})^{j_i}$, $\forall q_{j_i}^+ > q_{j_i}^*$, cannot belong to end-to-end shortest path.

Proof of Lemma 3: We prove by contradiction. Suppose shortest sub-path up to state $(q_{j_i}^+, p_{j_i})^{j_i}$, $q_{j_i}^+ > q_{j_i}^*$, is part of an end-to-end shortest path. If we replace sub-path to $(q_{j_i}^+, p_{j_i})^{j_i}$ with sub-path to $(q_{j_i}^*, p_{j_i})^{j_i}$, a synthesized view j' to the right of j_i and coded view j_{i+1} that depend on view j_i's texture map will have no larger distortion d_{j',j_i}^s or Lagrangian cost ϕ_{j_{i+1},j_i}, if $q_{j_i}^*$ is used instead of $q_{j_i}^+$, by monotonicity in quantization level (8) and (9). Given $\Phi_{j_i}(q_{j_i}^+, p_{j_i}) > \Phi_{j_i}(q_{j_i}^*, p_{j_i})$, we see that replacing sub-path to $(q_{j_i}^+, p_{j_i})^{j_i}$ with sub-path to $(q_{j_i}^*, p_{j_i})^{j_i}$ will yield strictly lower Lagrangian cost. A contradiction. □

Lemma 3 also holds true for depth level p_{j_i}: given q_{j_i}, if $\Phi_{j_i}(q_{j_i}, p_{j_i}^+) > \Phi_{j_i}(q_{j_i}, p_{j_i}^*)$, $\forall p_{j_i}^+ > p_{j_i}^*$, then states $(q_{j_i}, p_{j_i}^+)^{j_i}$'s, $\forall p_{j_i}^+ > p_{j_i}^*$, are sub-optimal and can be skipped.

The second lemma eliminates *sub-optimal edges* from state $(p_j, q_j)^j$ of view j to a state in further-away coded view k using monotonicity in predictor's distance.

Lemma 4: Suppose the optimal sub-path from start state s to state $(q_j, p_j)^j$ of view j does not use depth map of view j for view synthesis. If edge $(q_j, p_j)^j \rightarrow (q_k, p_k)^k$ also does not use depth map of view j, then edge $(q_j, p_j)^j \rightarrow (q_k, p_k)^k$ cannot belong to the end-to-end shortest path.

Proof of Lemma 4: We prove by contradiction. Suppose an optimal end-to-end path includes edge $(q_j, p_j)^j \rightarrow (q_k, p_k)^k$. Let x be the state in 3D trellis prior to state $(q_j, p_j)^j$ in the shortest sub-path from s to state $(q_j, p_j)^j$. Suppose we replace edges $x \rightarrow (q_j, p_j)^j \rightarrow (q_k, p_k)^k$ in shortest path with edge $x \rightarrow (q_k, p_k)^k$. By assumption, depth map of view j is not used, hence there are no uncoded (synthesized) views between node x and view j, and between view j and k. That means coded views between x and k can be assigned the same texture map quantization levels q's in 2D trellis of replacement edge $x \rightarrow (q_k, p_k)^k$, resulting in the same Lagrangian cost. Moreover, by not encoding depth map of view j, there is a non-zero cost saving for $\lambda > 0$. Hence a path using the replacement edge instead will yield lower cost. A contradiction. □

The corollary of Lemma 4 is that if the said condition holds, then edges $(q_j, p_j)^j \rightarrow (q_k^+, p_k^+)^k$, where q_k^+ and p_k^+ are levels larger than or equal to q_k and p_k respectively, also cannot belong to the end-to-end shortest path. The reason is that views between j and k in edge $(q_j, p_j)^j \rightarrow (q_k, p_k)^k$ that do not use depth map of view j for view synthesis will surely not use the same depth map of view j if the texture and/or depth map

```
1) Compute edge cost $s \rightarrow (q_i, p_i)^i$ for all states $(q_i, p_i)^i$ and store as $\Phi_i(q_i, p_i)$. Compute edge cost $s \rightarrow t$ and store as $\Phi_t$. Initialize $i = 1$.
2) For each $p_i$ of view $i$, find $q_i^*$ s.t. $\Phi_i(q_i^+, p_i) > \Phi_i(q_i^*, p_i)$, $\forall q_i^+ > q_i^*$. Eliminate states $(q_i^+, p_i)^i$'s from consideration.
3) For each $q_i$ of view $i$, find $p_i^*$ s.t. $\Phi_i(q_i, p_i^+) > \Phi_i(q_i, p_i^*)$, $\forall p_i^+ > p_i^*$. Eliminate states $(q_i, p_i^+)^i$'s from consideration.
4) For each survived state $(q_i, p_i)^i$ of view $i$, evaluate forward edges to states $(q_j, p_j)^j$'s for each view $j$, $j > i$, as follows.
   a) Initialize length-$P_{\max}$ vector $\mathbf{Q}_{\text{lim}}$ to $[Q_{\max}, \ldots, Q_{\max}]$.
   b) for $y = 1$ to $P_{\max}$,
      i) for $x = 1$ to $\mathbf{Q}_{\text{lim}}(y)$,
         A) Evaluate edge $(q_i, p_i)^i \rightarrow (x, y)^j$.
         B) If neither shortest sub-path to $(q_i, p_i)^i$ nor edge $(q_i, p_i)^i \rightarrow (x, y)^j$ uses depth map of view $i$ for view synthesis, update $\mathbf{Q}_{\text{lim}}(y')$, $y \leq y' \leq Q_{\max}$, to $x - 1$.
5) If $i < V$, increment $i$ by 1 and repeat step 2 to 4.
```

Fig. 5. Efficient Shortest Path Search in 3D Trellis

of predictor view k is of a coarser quality, by monotonicity in predictor's quantization level (9).

1) Efficient Algorithm for 3D Trellis: Given the two derived lemmas, we now describe an efficient shortest path search algorithm for 3D trellis, shown in Fig. 5. Starting from start state s, 3D edges to each state $(q_i, p_i)^i$ of view i are evaluated and stored in $\Phi_i(q_i, p_i)$ as initial values. Then for each view i, each state $(q_i^+, p_i)^i$, where $\Phi_i(q_i^+, p_i) > \Phi_i(q_i^*, p_i)$ and $q_i^+ > q_i^*$, is eliminated from shortest path consideration due to Lemma 3. Similar step is taken to eliminate $(q_i, p_i^+)^i$, where $\Phi_i(q_i, p_i^+) > \Phi_i(q_i, p_i^*)$ and $p_i^+ > p_i^*$.

In step 4, for each survived state $(q_i, p_i)^i$ of view i, we evaluate all forward sub-paths to states $(q_j, p_j)^j$'s of view j, but only if either shortest sub-path to $(q_i, p_i)^i$ or edge $(q_i, p_i)^i \rightarrow (q_j, p_j)^j$ uses depth map of i. If not, edges $(q_i, p_i)^i \rightarrow (q_j^+, p_j^+)^j$ are eliminated as well.

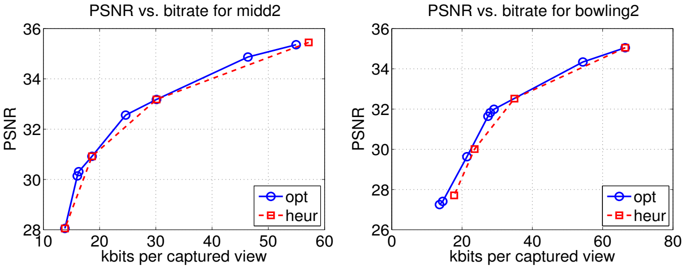
VI. Experimentation

A. Experimental Setup

To test the effectiveness of our proposed optimization scheme, we used H.264 JM16.2 video codec to encode texture and depth maps (texture and depth maps were encoded separately), and used ViSBD 2.1 as view synthesis tool at the receiver. For test sequences, we used two Middlebury still image sequences [9], `midd2` and `bowling2`, of size 1366×1110 and 1330×1110, respectively, with seven captured views each. We assumed the available quantization levels for both texture and depth maps were $\mathcal{Q} = \mathcal{P} = \{10, 15, \ldots, 50\}$. Rate controls were disabled in JM16.2, and software modifications were made so that a particular quantization level can be specified for each individual frame.

B. Experimental Results

We compare our proposed bit allocation & view synthesis algorithm (`opt`) to a heuristic scheme (`heur`) that selects texture maps of all views for encoding and assigns constant quantization levels for all encoded maps (applying straightforwardly the H.264 JM16.2 video codec to the captured texture maps). We see the performance of the two schemes in a plot of visual quality (Peak Signal-to-Noise Ratio (PSNR)) versus encoding rate per captured view in Fig. 6. We see that `opt` performed better than `heur` generally for all bitrate range, and in particular, outperformed `heur` by up to 0.80dB and 0.83dB for `midd2` and `bowling2` respectively at low bitrate.



(a) `midd2` (b) `bowling2`

Fig. 6. Performance Comparison between Optimal and Heuristic View and Quantization Level Selection Schemes

To estimate the computation savings we achieved using our proposed `opt` over a full 3D trellis search, we counted the number of times the cost of a 3D edge need to be calculated in the 3D trellis by `opt`. We found that `opt` saved up to 66% in computation over the full 3D trellis search.

VII. Conclusions

In this paper, we address the problem of how to best select texture and depth maps of captured views for encoding at appropriate quantization levels, such that the reconstruction fidelity of a designated set of V views at receiver is maximized for given bitrate constraint. We show that the optimal solution corresponds to the shortest path in a 3D trellis. We that derive a computation-efficient search algorithm, exploiting the monotonicity property, that finds the shortest path by visiting only a subset of nodes and edges in the trellis. We show that our scheme can reduce computation by up to 66% over the full trellis search, and can achieve up to 0.83dB gain in PSNR over a naive constant quantization scheme.

References

[1] M. Flierl, A. Mavlankar, and B. Girod, "Motion and disparity compensated coding for multiview video," in *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 17, no.11, November 2007, pp. 1474–1484.

[2] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," in *IEEE Trans. on Circuits and Systems for Video Tech.*, vol. 17, no.11, November 2007, pp. 1461–1473.

[3] P. Merkle, A. Smolic, K. Mueller, and T. Wiegand, "Multi-view video plus depth representation and coding," in *IEEE International Conference on Image Processing*, San Antonio, TX, October 2007.

[4] Y. Morvan, D. Farin, and P. H. de With, "Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images," in *IEEE International Conference on Image Processing*, San Antonio, TX, September 2007.

[5] M. Maitre, Y. Shinagawa, and M. Do, "Wavelet-based joint estimation and encoding of depth-image-based representations for free-viewpoint rendering," in *IEEE Transactions on Image Processing*, vol. 17, no.6, June 2008, pp. 946–957.

[6] Y. Liu, Q. Huang, S. Ma, D. Zhao, and W. Gao, "Joint video/depth rate allocation for 3D video coding based on view synthesis distortion model," in *Elsevier, Signal Processing: Image Communication*, vol. 24, no.8, September 2009, pp. 666–681.

[7] Y. Shoham and A. Gersho, "Efficient bit allocation for an aibitrary set of quantizers," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no.9, September 1988, pp. 1445–1453.

[8] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," in *IEEE Transactions on Image Processing*, vol. 3, no.5, September 1994.

[9] "2006 stereo datasets," http://vision.middlebury.edu/stereo/data/scenes2006/.