PRECISION ENHANCEMENT OF 3D SURFACES FROM MULTIPLE QUANTIZED DEPTH MAPS

Pengfei Wan[o], Gene Cheung[#], Philip A. Chou[$], Dinei Forencio[$], Cha Zhang[$], Oscar C. Au[o]

[o] Hong Kong University of Science and Technology, [#] National Institute of Informatics,
[$] Microsoft Research

ABSTRACT

Transmitting from sender compressed texture and depth maps of multiple viewpoints enables image synthesis at receiver from any intermediate virtual viewpoint via depth-image-based rendering (DIBR). We observe that quantized depth maps from different viewpoints of the same 3D scene constitutes multiple descriptions (MD) of the same signal, thus it is possible to reconstruct the 3D scene in higher precision at receiver when multiple depth maps are considered jointly. In this paper, we cast the precision enhancement of 3D surfaces from multiple quantized depth maps as a combinatorial optimization problem. First, we derive a lemma that allows us to increase the precision of a subset of 3D points with certainty, simply by discovering special intersections of quantization bins (QB) from both views. Then, we identify the most probable voxel-containing QB intersections using a shortest-path formulation. Experimental results show that our method can significantly increase the precision of decoded depth maps compared with standard decoding schemes.

Index Terms— Texture-plus-depth representation, 3D reconstruction, multiple descriptions

1. INTRODUCTION

Texture-plus-depth [1] has quickly become a popular format for dynamic 3D scene representation. One reason is because receiver can use texture and depth maps transmitted from sender from different viewpoints for synthesis of novel images as seen from freely chosen virtual viewpoints via *depth-image-based rendering* (DIBR) [2]. Another reason is because mature video coding tools like H.264 [3] and HEVC [4] can be easily and modestly adjusted for compression of the new video format. To reduce coding rate to reasonable size, however, input texture and depth videos are typically lossily compressed via quantization using these tools, resulting in quantization errors at decoder that corrupt the fidelity of the reconstructed signal.

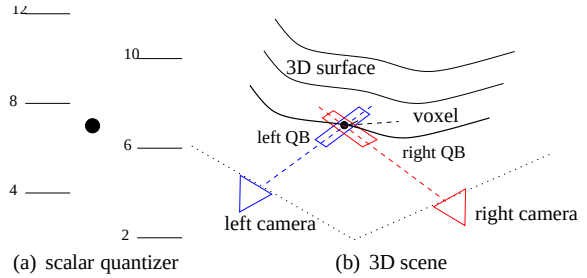


Fig. 1. Multiple descriptions for scalar quantizers and for 3D scene.

This work was supported by Microsoft Research CORE program.

To lessen the ill effects of depth video quantization at receiver, we observe that *quantized depth maps from different viewpoints of the same 3D scene constitute multiple descriptions (MD) of the same signal*. Thus, it is possible to enhance the depth precision of the described 3D scene at receiver when multiple quantized depth maps are considered jointly. As a motivating analogy, consider first MD of scalar quantizers; an example is shown in Fig. 1(a) where there are two scalar quantizers offset by 2 from each other. If the decoder receives only the quantization bin (QB) index of the left quantizer, then one can only deduce the coded scalar to be between 4 and 8. If the decoder receives QB indices of both left and right quantizers, then one can deduce the scalar to exist in the *intersection* of the two QBs—concluding the scalar to be between 6 and 8—enhancing the precision from 4 to 2.

Similarly, consider a 3D point (called *voxel* in the sequel) in the captured 3D scene that is visible from both left and right cameras, as shown in Fig. 1(b), resulting in one sample in each of the two depth maps[1]. If only the left depth sample is considered, then decoder can only deduce that a voxel exists inside the one QB (blue QB in Fig. 1(b)). If both depth samples are considered, then decoder can deduce that a voxel exists in the *intersection* of the two QBs, enhancing the depth resolution of the 3D scene.

Unlike the scalar quantizer case, however, correct matching of a pair of left and right QBs that contain the same voxel is non-trivial. In this paper, we formalize the QB matching problem to enhance depth precision of the 3D scene at receiver as a combinatorial optimization problem. First, we derive a lemma that allows us to increase the precision of a subset of 3D points with certainty, simply by discovering special intersections of quantization bins (QB) from both views. Then, we identify the most probable voxel-containing QB intersections using a shortest-path formulation. Experimental results show that our proposed method significantly outperforms single depth map in accuracy with respect to the ground truth signal.

The outline of the paper is as follows. We first discuss related work in Section 2. We then overview our system in Section 3. We formalize our optimization in Section 4. Finally, experimental results and conclusions are presented in Section 5 and 6, respectively.

2. RELATED WORK

While much efforts have been invested into efficient compression schemes for 3D visual data in texture-plus-depth format—e.g., unique characteristics of depth maps like piecewise smoothness have been exploited to improve depth map coding efficiency [5, 6]—majority of the proposals are simple extensions or modest modifications of existing coding tools like H.264 [3] or HEVC [4] instead of a complete coding architecture overhaul. It is thus likely that

[1]We will assume spatial resolutions of the depth maps are sufficiently high to provide enough samples for this assumption to hold true.

the same hybrid motion-compensation / transform-based residual coding framework will remain in place for the foreseeable future.

Nonetheless, we stress that our proposed depth precision enhancement algorithm is applicable to any texture-plus-depth coding scheme from which we can derive an independent QB for each depth sample in each view. For block-based coding schemes like H.264 where transform coefficients of a K-pixel block are quantized and transmitted, one can derive a QB for each depth pixel in the block as follows. We first identify the K-dimensional quantization region that corresponds to the scalar quantized coefficients of the K-pixel block. If the same quantization step size is used for each coefficient, then the quantization region is a hypercube. We then construct a *bounding box*[2] with sides that are either parallel or perpendicular to the pixel domain axes, that tightly contains the quantized region (solvable via linear programming). The width of the bounding box along each pixel domain axis is the size of the QB for that pixel.

While we focus our study of precision enhancement of 3D surface using quantized depth maps at the decoder only, knowledge gained from our study can be leveraged at the encoder, so that appropriate bit allocation can be performed among the multiple coded depth maps, improving overall rate-distortion (RD) performance. Joint optimization of depth map coding at encoder and depth map enhancement at decoder will be considered as future work.

3. SYSTEM OVERVIEW

(a) texture map (b) depth map

Fig. 2. Left view for the dude sequence.

We consider a scenario where the most likely 3D surface is estimated at the decoder, given quantized color / depth map pairs from two camera viewpoints (left and right). See Fig. 2 for an example. The color / depth map pairs are rectified, so that a row of pixels in the left view corresponds to a row of pixels in the right view. We assume that the depth maps are coarsely quantized compared to the spatial resolution. That means for each voxel on the 3D surface, there is both a color and a depth pixel sample in both the left and right view, *if* the voxel is visible from both viewpoints (no occlusion). Our depth resolution enhancement work is constructed based on this *double-sample* assumption. We further assume Lambertian surface characteristic for the 3D scene, so that the same voxel visible from both views will result in similar color (RGB) values.

3.1. QBs, ICs and Quantized Curves

Given the depth maps are rectified, we consider one row of pixels in the two depth / color map pairs at a time, corresponding to a 2D epipolar plane in 3D space. Possible depth values at each pixel location are partitioned into *quantization bins* (QB). The shape of a QB on the epipolar plane is approximated by a rectangle, whose width depends on the spatial resolution and length depends on the depth

[2]Though the size of the bounding box is larger than the hypercube, our depth enhancement algorithm can nonetheless improve depth precision of the decoded depth signal.

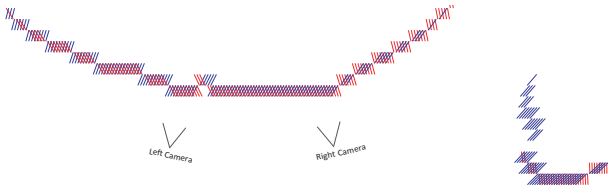


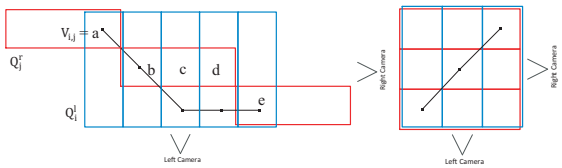
Fig. 3. One epipolar plane of dude's two views. Active QB is represented by a line segment, and the intersection of two active QBs is an IC.

quantization granularity. The QB with index that is actually coded is called *active*; the captured voxel of the 3D surface must exist within the active QB confine. The i-th QB from the left and right views are denoted as Q_i^l and Q_i^r, respectively.

A *cell* is an intersection of two QBs. We denote the intersection of i-th QB of left camera (Q_i^l) and j-th QB of right camera (Q_j^r) by $V_{i,j}$; see Fig. 4(a). We reserve the term *intersection cell (IC)* to mean intersection of two active QBs. Note that an active QB may have multiple ICs with different active QBs from another view.

An IC is called *true* if it contains a voxel that is part of the actual 3D surface. Since an IC is by definition smaller than QB in size (higher precision), the problem of depth precision enhancement is thus the selection of true ICs within active QBs.

On the epipolar plane, the 3D surface can be divided into individual contiguous *segments*; e.g., foreground and background segments. A *quantized curve* is a spatially contiguous series of QBs (at low precision) or ICs (at high precision). Fig. 3 shows QBs from a single pixel row in left and right views of the dude sequence.



(a) special case (b) general case

Fig. 4. Example for deterministic ICs and probabilistic ICs. Voxels (black dots) are connected to show the original 3D curve.

3.2. Deterministic ICs

We first identify ICs that can be certified as true with probability 1 (called *deterministic ICs*) without color information. We identify these ICs with the following lemma.

Lemma 1. *$V_{i,j}$ is true with probability 1 if: (i) $V_{i,j}$ is the only IC of QB Q_i^l and other cells of Q_i^l are not occluded by active QBs in right view; or (ii) $V_{i,j}$ is the only IC of Q_j^r and other cells of Q_j^r are not occluded by active QBs in the left view.*

As an example, there are five ICs (a to e) in Fig. 4(a), only a and e satisfy Lemma 1 and thus are deterministic ICs.

We outline a proof of Lemma 1 as follows. Suppose the first condition in Lemma 1 is true. Because other cells $V_{i,k}$ of Q_i^l are visible from the right camera (not occluded), there would be an active QB Q_k^r intersecting with active QB Q_i^l if there is a voxel in cell $V_{i,k}$. However, we know active QB Q_i^l only intersects with active QB Q_j^r. Hence, there can be no voxels in $V_{i,k}$, $k \neq j$. Since QB Q_i^l is active, $V_{i,j}$ must be true.

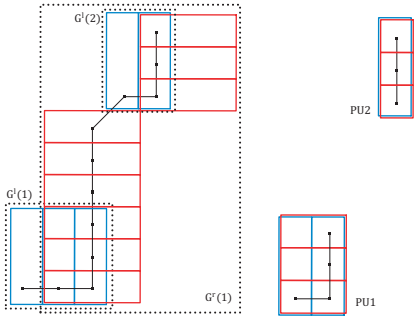
In general, it is possible that no cells in a local area satisfy the condition in Lemma 1; see Fig. 4(b) for an example. In this case, we

will use color information to disambiguate among candidate cells in a probabilistic manner.

4. QUANTIZED CURVE ESTIMATION

Having identified deterministic ICs, we now formulate the problem of estimating the most likely quantized curve. We first divide QBs on an epipolar plane into segments, so that contiguity of quantized curve can be enforced within a segment. Each segment is further divided into *process units* (PU), each with well defined start and end cells. Finally, we estimate a contiguous *maximum liklihood* (ML) quantized curve for each PU using shortest-path formulations.

4.1. Grouping QBs into Segments



(a) grouping into a segment (b) 2 PUs of (a)

Fig. 5. Grouping QBs into segments and dividing a segment into PUs.

To group QBs in an epipolar plane into segments, we do the following. First, neighboring active QBs of the left view—two QBs are neighbors if they are side-by-side or diagonal from each other—are grouped together as $\{G^l(k)\}$. The same procedure is performed for active QBs of the right view, resulting in $\{G^r(k)\}$. Then, for each pair of groups that have at least one overlapping cell, we take the union of them to be a new combined group. We continue this step until no more group pairs have overlapping cells; the remaining groups are the individual segments. As an example, in Fig. 5(a) all the left and right groups are merged into one segment because they have overlapping cells. A segment represents an actual physical object in the 3D scene, e.g. a person's body. Hence we will enforce contiguity within a segment when estimating a quantized curve.

4.2. Dividing Segment into Process Units

We now identify one or more PUs in a segment. A PU is composed of ICs only. In the following sections, we estimate an ML quantized curve for each PU independently. For any *indeterminant QB*—an active QB with at least one non-IC cell (i.e. color information is available from only one view)—that connects PUs into a segment, we will choose the middle cell for curve reconstruction to minimize worst-case error, as done in conventional decoding schemes.

We first search for the left-most active vertical QB containing ICs from the left view (second blue column from the bottom-left in Fig. 5(a)); these are the first ICs in the first PU. We initialize the middle cell in the QB to the left of this QB as start cell V_s.

For each side-by-side vertical QB to the right that contains ICs, we add the corresponding ICs to the PU; see the 3×2 ICs in the bottom left of Fig. 5(a). At the right-most QB of this PU, by segment construction there are only two cases: i) an active vertical QB diagonal from this PU, or ii) an active horizontal QB on top of this PU (shown in Fig. 5(a)). In the first case, the corner cell that connects to the diagonal QB is the end cell V_e of this PU. The connecting corner cell of the diagonal QB is the start cell of the new PU, if the QB is not indeterminant. If it is, then the middle cell is selected.

In the second case, the horizontal QB on top must be indeterminant, and so we pick the middle cell as the end cell V_e of the first PU. At the top of this heap of horizontal QBs, by segment construction there will be a diagonal cell. If this cell belongs to an active vertical QB, then the situation is same as case one above. If not, it must belong to an active horizontal QB, and the situation is then the same as case two above. This procedure is repeated until all the cells in the segment are examined.

In the end, one or more PUs with corresponding start and end cells V_s and V_e are identified within a segment. Note that some of the V_s (V_e) are not ICs, which will be addressed in Section 4.4.

4.3. Maximum Likelihood Formulation

For a given PU, we now formulate the IC selection problem in a ML formulation. We first construct a graph $\mathcal{G}$: each IC $V_{i,j}$ is a node that is connected to its neighboring ICs $\mathcal{N}_{i,j}$—to the left, right, top, down and diagonal—with edges[3]. Given color information from the left and right views, $\{\mathbf{Y}^l, \mathbf{Y}^r\}$, our goal is to find the ML quantized curve—a most likely *ordered* set of nodes denoted by $\mathbf{C} = \{V^1, \ldots, V^K\}, V^k \in \mathcal{G}$, of some size K:

$$\max_{\mathbf{C}} Pr(\mathbf{Y}^l \mathbf{Y}^r | \mathbf{C}), \quad \text{s.t. } \mathbf{C} \in \mathcal{C} \tag{1}$$

where $\mathcal{C}$ is the feasible set of quantized curves in a PU. Any $\mathbf{C} \in \mathcal{C}$ must satisfy the following constraints:

1. $\forall Q_i^l,\ \exists V_{i,n} \in \mathbf{C}$ for some n.
2. $\forall Q_j^r,\ \exists V_{m,j} \in \mathbf{C}$ for some m.
3. $\forall V_{i,j}^k \in \mathbf{C}, 1 < k < K, \exists V^{k-1}, V^{k+1} \in \mathcal{N}_{i,j}$.

Constraints 1 and 2 state that a feasible curve must include at least one IC in each active QB. Constraint 3 states that a feasible curve must be contiguous within a PU.

Probabilities of elements in $\mathbf{C}$ are assumed independent. Using color matching as conditional probability, (1) becomes:

$$\begin{aligned} &\max_{\mathbf{C} \in \mathcal{C}} \prod_{k=1}^{K} Pr(\mathbf{Y}_k^l \mathbf{Y}_k^r | V^k) \\ \Leftrightarrow &\min_{\mathbf{C} \in \mathcal{C}} \sum_{k=1}^{K} -\log Pr(\mathbf{Y}_k^l \mathbf{Y}_k^r | V^k) \end{aligned} \tag{2}$$

Note that (2) is essentially a sum of node costs (or edge weights) along a contiguous curve $\mathbf{C}$. Specifically, the weight of an edge arriving at V^k is $W_k = -\log Pr(\mathbf{Y}_k^l \mathbf{Y}_k^r | V^k)$, i.e. the consistency (color matching) of V^k's color vectors from the two views ($\mathbf{Y}_k^l$ and $\mathbf{Y}_k^r$). For example, $W_k = ||\mathbf{Y}_k^l - \mathbf{Y}_k^r||_1$ if we assume Laplacian probability model for color matching. Exceptions are made for deterministic ICs as arriving nodes with edge weights $W = 0$.

4.4. Shortest Path as Estimated Quantized Curve

Given graph $\mathcal{G}$ and start and end cells V_s and V_e for each PU, we argue that a suitable variant of a shortest-path formulation will result in an ML optimal solution for defined feasible set $\mathcal{C}$. We start from the simplest case where V_s and V_e are opposite corner ICs of the PU; an example is PU2 in Fig. 6. In this case, $\mathcal{C}$ is the set of all paths

[3]We connect a middle cell of an indeterminant QB to its neighboring ICs in the same way, but if none exists, we draw a single edge to its nearest IC.

between V_s and V_e. The solution of (2) is then simply the shortest path from V_s to V_e on $\mathcal{G}$. This can be solved efficiently using any shortest path algorithms, such as Bellman-Ford (BF) [7].

If V_s and V_e are corner ICs on the same side (e.g. PU3), feasible solution set $\mathcal{C}$ is the set of paths from V_s to V_e that must pass through at least one *intermediate cell* V_i of the furthest row or column (the intermediate V_i for PU3 is the three ICs in the last row). This can be solved by calling BF twice (with starting node fix at V_s or V_e for each run), and choosing the union of two shortest paths $V_s \rightarrow V_i$ and $V_i \rightarrow V_e$, whose sum of costs is minimal among all possible V_i.

When V_s or V_e is not an IC (e.g. PU1), the start (end) cells of $\mathcal{G}$ become the ICs that are connected to V_s or V_e (which is outside the PU). Because V_s (V_e) has at most 3 connected ICs in the PU, the number of combinations of start and points for the PU is at most 9. For each combination, we need to assign at most 2 intermediate cells to satisfy constraints 1 and 2 (at least one IC of an active QB must be traversed). This can be similarly solved by calling BF multiple times.

As a whole, the ML quantized curve can be calculated in polynomial time for any PU. Combining the ML quantized curves for all PUs in all segments on all epipolar planes, we arrive at a *quantized 3D surface* with reduced uncertainty (enhanced precision).

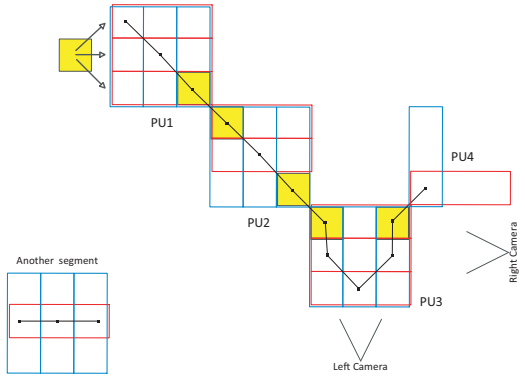


Fig. 6. Example of PUs with different start/end cells (marked in yellow).

5. EXPERIMENTATION

Two test sequences, `sphere` (400×400) and `dude` (480×800), are used for experiments. They are both composed of two rectified views with depth and color maps. In `sphere`, the camera baseline is 5.4 and the depth range is $[9.0, 10.16]$; in `dude`, the camera baseline is 1.0 and the depth range is $[1.15, 2.6]$.

To decode depth values, the standard method picks the center depth values of QBs. In our method, center values of the estimated quantized 3D surface (composed of ICs in the solution of (2), and middle cells for indeterminant QBs) are used as decoded depth.

Mean Square Error (MSE) ε is used as metric:

$$\begin{aligned} \varepsilon &= (\varepsilon^l + \varepsilon^r)/2 \\ \varepsilon^l &= \frac{1}{MN}||\mathbf{D}^l - \hat{\mathbf{D}}^l||_F^2, \quad \varepsilon^r = \frac{1}{MN}||\mathbf{D}^r - \hat{\mathbf{D}}^r||_F^2 \end{aligned} \tag{3}$$

where ε^l and ε^r are respectively the MSE of the left and right decoded depth maps. $\hat{\mathbf{D}}$ is the ground-truth 12-bit depth map. $\mathbf{D}$ is the decoded depth map. $M \times N$ is the spatial resolution of the sequence.

Depth maps with varying bit-precision (3-bit$\sim$6-bit, denoted by d3$\sim$d6 respectively) are used as inputs. Color maps with 6-bit or 8-bit precision (c6 and c8) are used as side information. MSE results are shown in Table. 1, where 'sta' refers to standard method and 'our' refers to proposed method.

Table 1. MSE Comparisons

	`sphere`			`dude`		
	sta	our-c6	our-c8	sta	our-c6	our-c8
d3	2.30e-3	1.96e-4	1.66e-4	4.28e-3	1.07e-3	1.07e-3
d4	5.02e-4	8.55e-5	5.82e-5	7.73e-4	2.00e-4	2.00e-4
d5	1.18e-4	4.61e-5	2.51e-5	1.86e-4	5.36e-5	5.36e-5
d6	2.86e-5	2.38e-5	1.37e-5	4.28e-5	1.69e-5	1.69e-5

We can see that our method is able to achieve significantly higher precision: the MSE of proposed method is less than 10% of that of standard method for `sphere` with 3-bit input depth and 6-bit color. Although in general lower MSE will be obtained with better color information, 6-bit and 8-bit color maps didn't make a difference for `dude` whose color tends to be locally uniform; see Fig. 2.

Some visual results are shown in Fig. 7. We can see that our solution aligns with ground-truth 3D surface much better than standard method who simply picks the center of QBs.

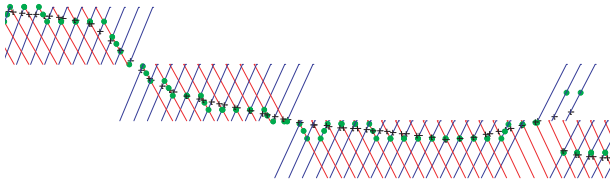


Fig. 7. Example of decoded surface of proposed method (green spots) and ground-truth (black crosses) for `dude` with 6-bit depth and 6-bit color.

6. CONCLUSION

In this paper, we consider the scenario of recovering a high precision 3D surface represented by multi-view texture-plus-depth maps. We formulate it as a maximum likelihood problem which can be effectively solved using a shortest-path algorithm. Effectiveness of proposed method is verified in accuracy of decoded depth maps.

7. REFERENCES

[1] 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.

[2] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3D video," in *App. of Digital Image Processing XXXII, Proceedings of the SPIE*, 2009, vol. 7443 (2009), pp. 74430T–74430T–11.

[3] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," in *IEEE Trans. on CSVT*, July 2003, vol. 13, no.7, pp. 560–576.

[4] G. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," in *IEEE Transactions on CSVT*, December 2012, vol. 22, no.12, pp. 1649–1668.

[5] A. Sanchez, G. Shen, and A. Ortega, "Edge-preserving depth-map coding using graph-based wavelets," in *43th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, November 2009.

[6] P. Merkle, C. Bartnik, K. Muller, D. Marpe, and T. Weigand, "3D video: Depth coding based on inter-component prediction of block partitions," in *2010 Picture Coding Symposium*, Krakow, Poland, May 2012.

[7] Cormen, Leiserson, and Rivest, *Introduction to Algorithms*, McGraw Hill, 1986.