Rate-constrained 3D Surface Estimation from Noise-corrupted Multiview Depth Videos

Wenxiu Sun *Member, IEEE*, Gene Cheung *Senior Member, IEEE*, Philip A. Chou *Fellow, IEEE*, Dinei Florencio *Senior Member, IEEE*, Cha Zhang *Senior Member, IEEE*, Oscar C. Au *Fellow, IEEE*

Abstract—**Transmitting compactly represented geometry of a dynamic 3D scene from a sender can enable a multitude of imaging functionalities at a receiver, such as synthesis of virtual images at freely chosen viewpoints via depth-image-based rendering (DIBR). While depth maps—projections of 3D geometry onto 2D image planes at chosen camera viewpoints—can nowadays be readily captured by inexpensive depth sensors, they are often corrupted by non-negligible acquisition noise. Given depth maps need to be denoised and compressed at the encoder for efficient network transmission to the decoder, in this paper we consider the denoising and compression problems jointly, arguing that doing so will result in a better overall performance than the alternative of solving the two problems separately in two stages. Specifically, we formulate a rate-constrained estimation problem, where given a set of observed noise-corrupted depth maps, the most probable (maximum a posteriori (MAP)) 3D surface is sought within a search space of surfaces with representation size no larger than a pre-specified rate constraint. Our rate-constrained MAP solution reduces to the conventional unconstrained MAP 3D surface reconstruction solution if the rate constraint is loose. To solve our posed rate-constrained estimation problem, we propose an iterative algorithm, where in each iteration the structure (object boundaries) and the texture (surfaces within the object boundaries) of the depth maps are optimized alternately. Using the MVC codec for compression of multiview depth video and MPEG free viewpoint video sequences as input, experimental results show that rate-constrained estimated 3D surfaces computed by our algorithm can reduce coding rate of depth maps by up to 32% compared to unconstrained estimated surfaces for the same quality of synthesized virtual views at the decoder.**

Index Terms—**multiview video, image denoising, depth image compression**

I. Introduction

With the advent of consumer-level depth capturing sensors [1] like Microsoft Kinect, depth images (per-pixel distances between objects in the 3D scene and the capturing camera) can now be acquired cheaply from multiple viewpoints. Each depth map constitutes a projection of the 3D geometry in the scene to a 2D image of fixed resolution. Thus, having acquired depth maps from multiple camera viewpoints, one can back-project them to the 3D space to (partially) recover the original 3D geometry. If the multiview depth maps—a representation of the 3D geometry—are compressed and transmitted, then the receiver can perform a range of 3D imaging tasks, such as synthesis of virtual images from freely chosen viewpoints using texture and depth maps of neighboring camera views via depth-image-based rendering (DIBR) [2].

To enable high quality communication of 3D geometry from sender to receiver, however, we are faced with two practical problems. The first problem is to estimate the actual 3D geometry of the scene from the depth maps acquired from consumer-level depth sensors, which are typically corrupted by non-negligible acquisition noise. The second problem is to find a compact representation for the estimated 3D geometry—one that does not require too many encoding bits—so that the communication cost will not be prohibitively high.

The conventional approach to these two problems—estimation of 3D surface from noisy observations and coding of chosen surface representation—is to treat these problems as independent and solve them separately one after another. For example, one can use a 3D surface reconstruction solution from the computer vision literature [3]–[5] first to derive the most probable 3D surface from noisy depth observations, then project this surface to chosen camera viewpoints as depth maps for compression. We argue that this is a sub-optimal approach; our concerned problem of identifying a surface representation that is *both* compact (require few encoding bits) *and* agrees with observations is inherently a *probabilistic* one. If one first computes a most probable surface with no consideration for representation size, and then sends it as a *deterministic* input to a lossy compression algorithm for depth map projection and coding, then all the probabilistic information obtainable from observed data that could potentially be useful for compression is lost. For example, a codec will not be able to compress lossily one part of the signal more aggressively than another, even if they have very different local noise statistics. (This argument will be presented more thoroughly in Section II.) See Fig. 1 for an illustration.

Wenxiu Sun and Oscar C. Au are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong (email: {eeshine, eeau}@ust.hk).

Gene Cheung is with National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo, 101-8430 Japan (email: cheung@nii.ac.jp).

Philip A. Chou, Dinei Florencio and Cha Zhang are with Microsoft Research, One Microsoft Way, Redmond, WA 98052 USA (email: {pachou, dinei, chazhang}@microsoft.com).

This work is supported by Microsoft Research CORE program.

In this paper, we instead address both aforementioned problems simultaneously by formulating a *rate-constrained estimation* problem: given a sequence of observed noise-corrupted depth maps $\mathbf{y}^t$, we seek the most probable (*maximum a posteriori* (MAP)) sequence of 3D surfaces $\hat{\mathbf{s}}^t$ *within* a search space of surfaces with representation size no larger than a pre-specified rate constraint, e.g., $r(\hat{\mathbf{s}}^t) \leq \bar{R}$. The most probable sequence of rate-constrained surfaces is then encoded using a MVC codec [6]. Our rate-constrained MAP solution reduces to an unconstrained MAP 3D surface reconstruction solution with no consideration for representation size if the rate constraint is loose. To formally formulate the problem, we first define an error term $-\log p(\hat{\mathbf{s}}^t|\mathbf{y}^t)$ that reflects the distance between the reconstructed 3D surface $\hat{\mathbf{s}}^t$ and the observed depth data $\mathbf{y}^t$. We then define a rate term that estimates the coding bits of the re-projected depth maps $\mathbf{d}^t$ (from the computed most probable rate-constrained 3D surface $\hat{\mathbf{s}}^t$) given a MVC codec is used for coding. To solve this optimization problem, we propose an efficient algorithm that finds a locally optimal solution by iterating between two steps: i) align edges in depth maps of consecutive views to match *scene structure* across views; and ii) smooth surfaces within depth edges to match *scene texture* across views. Using the MVC codec [6] for compression of multi-view depth video and MPEG free viewpoint video test sequences as input, experimental results show that optimized 3D reconstructions computed by our algorithm can reduce coding rate of depth maps by up to 32% compared to unconstrained estimated surfaces for the same quality of synthesized virtual views at the decoder.

The outline of the paper is as follows. We first discuss related work in Section II. We then provide an overview of our system model in Section III. We formulate our rate-constrained estimation problem in Section IV. In Section V, we show how an alternative formulation of the problem can lead to a rate-distortion (RD) interpretation with additional insights. We discuss our optimization algorithm in Section VI. Experimental results and concluding remarks are presented in Sections VII and VIII, respectively.

II. Related Work

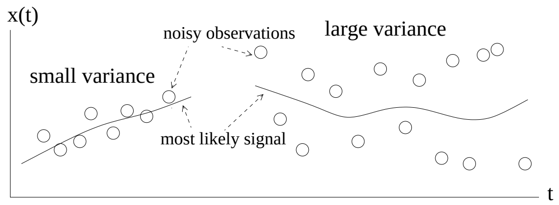


Fig. 1. Example of signal $x(t)$ with different noise variances at different spatial regions t, rendering a separate denoising / compression approach sub-optimal.

The problem of denoising depth observations has been studied extensively in the literature, and can be broadly divided into two categories: i) denoising of single depth images, and ii) reconstruction of 3D surfaces in space given noisy depth observations from multiple viewpoints. The recent advance of depth sensing technologies such as time-of-flight (ToF) cameras (e.g., Mesa Imaging SR4000) and structured light cameras (e.g., Microsoft first generation Kinect) has driven strong interest in the image processing community to study the depth image denoising problem [7]–[14]. A common thread to these works is the exploitation of the known piecewise smooth signal prior in depth images for denoising. A large portion of this work further assume the availability of a perfectly aligned color image along with the depth image as side information for depth denoising [7]–[13]. Some further assume the unique noise characteristics of depth images captured by structured light cameras [11]–[13], which are very different from ToF cameras. Though we also exploit the piecewise smooth characteristic of depth images in our algorithm, we differ in that we jointly solve the depth image denoising and compression problem at the same time via a rate-constrained surface estimation formulation to be discussed in Section IV.

In parallel, the problem of reconstructing a 3D surface given noise-corrupted depth observations from multiple viewpoints has been studied extensively in the computer vision literature [3]–[5]. It can be argued that algorithms for solving this problem produce surfaces that are, at least approximately, most likely given the observations. However, even reconstructed 3D surfaces that are optimal in this sense may require a large encoding overhead. One naïve approach to finding a good rate-constrained 3D surface is to separate the problem into two steps: i) first estimate the underlying (ground truth) 3D surface from noise-corrupted observations regardless of representation size; and then ii) perform conventional RD optimization as done in a standard video codec like H.264 [15] given the estimated signal as input. We argue that this is a sub-optimal approach. The problem of finding an optimal rate-constrained 3D surface from noise-corrupted observations is inherently a *probabilistic* one: identifying the most likely 3D surface (one that maximizes the posterior probability) *within* a set of surfaces $\hat{\mathbf{s}}$ with representation size no larger than a bit budget, i.e. $r(\hat{\mathbf{s}}) \leq \bar{R}$. By first identifying the most likely 3D surface given observations and then performing RD optimization on this estimate to arrive at a surface $\hat{\mathbf{s}}$ with rate $r(\hat{\mathbf{s}})$ not exceeding budget $\bar{R}$, there is no guarantee that the computed $\hat{\mathbf{s}}$ is indeed the most likely one in the rate-constrained space $\hat{\mathcal{S}}(\bar{R}) = \{\, \hat{\mathbf{s}} \mid r(\hat{\mathbf{s}}) \leq \bar{R} \,\}$.

As an illustration, consider the 1D signal example in Fig. 1. In the first part of the signal, the local noise variance is smaller than in the second part. If a separate denoising / compression approach is taken, then a most probable signal $\mathbf{s}^*$ is first constructed, and then used as a deterministic input to a signal encoder to compute an approximate surface $\mathbf{s}^\#$ to most likely $\mathbf{s}^*$, such that $r(\mathbf{s}^\#) \leq \bar{R}$. However, doing so would mean that the information about the magnitude of local noise variance will be lost, and the first and second part of the signal would be lossily compressed equally. In contrast, a joint denoising / compression approach will observe that the noise variance of the second part is larger, and so the second part can be more aggressively compressed, given that the uncertainty in the estimated signal in the second part is larger, resulting in another surface $\mathbf{s}^o$, $r(\mathbf{s}^o) \leq \bar{R}$. In other words, though both surfaces $\mathbf{s}^\#$ and $\mathbf{s}^o$ are in the feasible space $\hat{\mathcal{S}}(\bar{R})$, $\mathbf{s}^o$ is the more probable one, resulting in performance gain. We will demonstrate empirically that our rate-constrained estimated 3D surfaces indeed outperform surfaces generated by this separation approach in Section VII.

In previous multiview depth map compression work, it has been observed that inconsistency among input depth maps of different views due to acquisition noise incurs expensive coding overhead, but does not lead to better synthesized view quality. Thus, denoising methods to improve inter-view consistency have been proposed [16], [17]. Our work is fundamentally different in that we seek a *most probable 3D surface* in a search space of rate-constrained surfaces, where the chosen surface is then projected to a number of camera viewpoints for compact representation as compressed multiview depth video. Thus, by construction our input depth maps to a multiview codec are always inter-view consistent. Furthermore, the set of generated consistent depth maps represents not just any 3D surface, but one that is most probable within the feasible space as dictated by the rate constraint.

III. System Overview

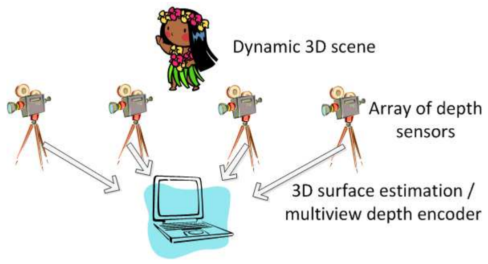


Fig. 2. Overview of multiview depth capturing system for a dynamic 3D scene.

We now provide an overview of our system model. We assume an array of V depth sensors capture depth images of the same dynamic 3D scene periodically from V different viewpoints, as shown in Figure 2. We assume the cameras have the same spatial resolution and are synchronized in time, as in [18]. The captured depth observations are corrupted by non-negligible acquisition noise, modeled as multivariate Gaussian. Given the observed depth data, the encoder first estimates a rate-constrained 3D surface of the scene, for a given bit budget of $\bar{R}$ bits per frame. The chosen 3D surface is then re-projected back to the camera views, which are subsequently encoded as multiview depth videos as a representation of the chosen 3D surface, using a known multiview video coding scheme like MVC [6], for transmission over a communication channel of bandwidth $\bar{R}$. The challenge is to estimate the most likely 3D surface given the observed depth data, subject to a representation size constraint $\bar{R}$. We discuss the formulation of this problem next.

IV. Problem Formulation

We now present our formulation of the rate-constrained 3D surface estimation problem. As a convention, matrices and vectors will be denoted respectively by boldface uppercase letters (e.g, $\mathbf{D}$) and lowercase letters (e.g., $\mathbf{d}$), and scalars will be denoted by italic upper or lowercase letters (e.g., n or N). Sets will be denoted by calligraphic letters (e.g., $\mathcal{S}$ or $\mathcal{Y}$).

Suppose one or more objects move freely in 3D space and are captured at each time instant t by a set of V depth cameras from different viewpoints, producing at each instant a set of observed depth maps $\mathbf{y}^t = \{\mathbf{y}_1^t, ..., \mathbf{y}_V^t\}$. We take $\mathbf{y}^t \in \mathcal{Y}$ to be an element of a real finite dimensional space $\mathcal{Y}$. Let $\mathbf{s}^t \in \mathcal{S}$ denote the underlying (i.e., ground truth not directly observed) surface of the object at instant t. One can think of $\mathcal{S}$ as the set of all surfaces, or 2D manifolds, in 3D. However, to avoid mathematical irregularities, we shall assume that the surfaces in $\mathcal{S}$ are "band-limited" in the sense of being describable with a finite number of parameters, e.g., as the limit of a subdivision surface [19]. Thus, we take $\mathbf{s} \in \mathcal{S}$ to be an element of a real finite dimensional parameter space $\mathcal{S}$.

Unlike previous work on 3D reconstruction from multiview depth data [3]–[5], we take a *rate-constrained estimation* approach. That is, we formulate our objective as finding the surface $\hat{\mathbf{s}}^t$ that maximizes the posterior probability density $p(\hat{\mathbf{s}}^t|\mathbf{y}^t)$ of the surface given the observations, or equivalently minimizes $-\log p(\hat{\mathbf{s}}^t|\mathbf{y}^t)$, subject to a constraint on the number of bits $r(\hat{\mathbf{s}}^t)$ used to encode a representation of $\hat{\mathbf{s}}^t$; i.e.,

$$\min_{\hat{\mathbf{s}}^t \in \hat{\mathcal{S}}} -\log p(\hat{\mathbf{s}}^t|\mathbf{y}^t) \qquad \text{s.t.} \quad r(\hat{\mathbf{s}}^t) \leq \bar{R} \tag{1}$$

Here, $\hat{\mathcal{S}} \subseteq \mathcal{S}$ is the codebook of all possible reproductions of surfaces $\hat{\mathbf{s}}^t$ by a given decoder, and $r(\hat{\mathbf{s}}^t)$ is the number of bits used to represent $\hat{\mathbf{s}}^t$ by a corresponding encoder. Thus, this is a rate-constrained *maximum aposteriori* (MAP) problem. It can be shown [20], [21], that if one is interested only in surfaces $\hat{\mathbf{s}}^t$ on the lower convex hull of the set $\{(r(\hat{\mathbf{s}}^t), -\log p(\hat{\mathbf{s}}^t|\mathbf{y}^t))\}$, this is equivalent to finding a $\hat{\mathbf{s}}^t$ that achieves

$$\min_{\hat{\mathbf{s}}^t \in \hat{\mathcal{S}}} -\log p(\hat{\mathbf{s}}^t|\mathbf{y}^t) + \lambda r(\hat{\mathbf{s}}^t) \tag{2}$$

for some Lagrange multiplier $\lambda > 0$.

By constraining the feasible search space of surfaces during estimation, *we are essentially solving the surface estimation problem and the compression problem at the same time*. As seen clearly in (1) and (2), the problem is inherently a joint one that cannot be separated into first an estimation sub-problem and then a compression sub-problem without sacrificing optimality. Thus, the joint estimation-compression approach can lead to better performance than a two-step separate approach. We will demonstrate this is indeed the case in our experiments in Section VII.

We refer to $-\log p(\hat{\mathbf{s}}^t|\mathbf{y}^t)$ as the error term and $r(\hat{\mathbf{s}}^t)$ as the rate term in (2). The error term measures, in some sense, the distance between the estimated surface $\hat{\mathbf{s}}^t$ and the observations $\mathbf{y}^t$. Note that the error term does not measure the distance between $\hat{\mathbf{s}}^t$ and the ground truth surface $\mathbf{s}^t$, since we have no direct observation of $\mathbf{s}^t$. Further, it does not measure the distance between $\mathbf{y}^t$ and $\hat{\mathbf{y}}^t$ — depth maps projected to the same V camera locations using reconstructed surface $\hat{\mathbf{s}}^t$ — since our ultimate goal is to reconstruct the underlying surface $\mathbf{s}^t$ rather than simply denoising observations $\mathbf{y}^t$. We can only hope to find a reconstruction $\hat{\mathbf{s}}^t$ that is somehow close to (i.e., explains) the observations $\mathbf{y}^t$, while satisfying one or more signal priors. It also means that we are free to use any form of compact representation for surface $\hat{\mathbf{s}}^t$ for coding, resulting in rate $r(\hat{\mathbf{s}}^t)$. For example, one can choose a different set of camera viewpoint locations than the observed V locations for projections into depth maps as representation [22].

Next, we will specify the error term using an assumed noise model and a set of signal priors. We then specify the rate term using proxies that approximate the coding rates of a typical multiview codec like MVC [6].

A. Error Term

We start by re-writing the posterior density $p(\mathbf{s}^t|\mathbf{y}^t)$ using Bayes' rule:

$$p(\mathbf{s}^t|\mathbf{y}^t) = \frac{p(\mathbf{y}^t|\mathbf{s}^t)\,p(\mathbf{s}^t)}{p(\mathbf{y}^t)}. \tag{3}$$

The numerator on the right side is a product of the likelihood $p(\mathbf{y}^t|\mathbf{s}^t)$ and the prior $p(\mathbf{s}^t)$. The likelihood $p(\mathbf{y}^t|\mathbf{s}^t)$ is modeled by the physics of the depth sensors and acquisition process as follows. At each instant t, first the underlying surface $\mathbf{s}^t$ is projected (with hidden surface removal) onto each of the V views, producing ideal depth maps $\mathbf{d}_1^t = \mathbf{d}_1(\mathbf{s}^t), \ldots, \mathbf{d}_V^t = \mathbf{d}_V(\mathbf{s}^t)$, which are deterministic functions of $\mathbf{s}^t$. From these ideal depth maps, the observed depth maps $\mathbf{y}_1^t, \ldots, \mathbf{y}_V^t$ are generated probabilistically according to a zero-mean Gaussian noise with conditional probability density

$$f(\mathbf{y}_v^t|\mathbf{d}_v^t) = \frac{|\mathbf{Q}_v^t|^{\frac{1}{2}}}{(2\pi)^{\frac{MN}{2}}} \exp\left(-\frac{1}{2}(\mathbf{y}_v^t - \mathbf{d}_v^t)^T\,\mathbf{Q}_v^t\,(\mathbf{y}_v^t - \mathbf{d}_v^t)\right), \tag{4}$$

where $\mathbf{Q}_v^t$ is the *precision matrix* (inverse of the covariance matrix) for the $M \times N$ depth map from camera v at instant t, and may depend on v, t, and even on the signal $\mathbf{d}_v^t$. For simplicity, we assume that the depth sensors are independent from each other and that the measurements are independent across time. This model can reasonably accommodate depth sensors based on stereo, structured light, or time-of-flight by accurately modeling the precision matrix [1], [23]. We can thus write likelihood $p(\mathbf{y}^t|\mathbf{s}^t)$ simply as:

$$p(\mathbf{y}^t|\mathbf{s}^t) = \prod_v f(\mathbf{y}_v^t|\mathbf{d}_v(\mathbf{s}^t)). \tag{5}$$

We next discuss the prior $p(\mathbf{s})$. In [24], if a 1D signal[1] $\mathbf{d}$ is assumed to be *piecewise linear*, then a prior term can be derived as follows. We first define matrix $\mathbf{D}$ as a one-sample right-shift operator on an input vector $\mathbf{d}$. For example, if the vector length of $\mathbf{d}$ is 5, then $\mathbf{D}$ is defined as:

$$\mathbf{D} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{6}$$

$\mathbf{D}\,\mathbf{d}$ then shifts the sample positions of $\mathbf{d}$ by one to the right. We can now write the prior density $g(\mathbf{d})$ of $\mathbf{d}$ as a Gaussian probability distribution follows:

$$g(\mathbf{d}) = \frac{1}{\rho}\exp\left(-\frac{\gamma}{2}\sum_{n=1}^{N}\left[\mathbf{d} - \frac{\mathbf{D}^n\mathbf{d} + \mathbf{D}^{-n}\mathbf{d}}{2}\right]^T \mathbf{W}(n)\left[\mathbf{d} - \frac{\mathbf{D}^n\mathbf{d} + \mathbf{D}^{-n}\mathbf{d}}{2}\right]\right) \tag{7}$$

where $\mathbf{D}^n$ now implies a right-shift of n samples, and $\mathbf{D}^{-n}$ implies a left-shift of n samples. Hence the term $\mathbf{d} - \frac{\mathbf{D}^n\mathbf{d}+\mathbf{D}^{-n}\mathbf{d}}{2}$ defines the *change in local gradient* in $\mathbf{d}$. For example, when $n = 1$, $(\mathbf{D}\,\mathbf{d} + \mathbf{D}^{-1}\,\mathbf{d})/2$ is the vector containing means of two immediate neighbors for each pixel, so if $\mathbf{d}$ is linear, the difference from $\mathbf{d}$ is zero.

$\mathbf{W}(n)$ is a diagonal weight matrix that penalizes the difference in local gradients. In general, the larger n is, the smaller the weight will be. ρ and γ are constants. We will discuss how $\mathbf{W}(n)$ can be defined in Section VI.

Since it is widely accepted that a depth map is piecewise smooth, we will use this function $g(\mathbf{d}_v^t)$ as our prior probability for each projected depth map $\mathbf{d}_v^t$ of view v, and $p(\mathbf{s}^t)$ will simply be:

$$p(\mathbf{s}^t) = \prod_v g(\mathbf{d}_v(\mathbf{s}^t))/Z, \tag{8}$$

where Z is a normalizing constant.

Combining (3), (5), and (8), we can now write the error term $-\log p(\hat{\mathbf{s}}^t|\mathbf{y}^t)$ as (9), where $K(\mathbf{y}^t)$ is a constant depending on $\mathbf{y}^t$, and $\mathbf{d}_1^t = \mathbf{d}_1(\mathbf{s}^t), \ldots, \mathbf{d}_V^t = \mathbf{d}_V(\mathbf{s}^t)$.

B. Rate Term

The rate term $r(\hat{\mathbf{s}})$ is the number of bits needed to signal to the decoder which surface $\hat{\mathbf{s}} \in \hat{\mathcal{S}}$ it should reproduce. In practice, we will use a decoder based on an existing multiview codec such as MVC, combined with a post-processing step to turn its decoded depth maps into a consistent 3D surface. This combination determines the set of all possible valid bit strings $\mathcal{C} = \{\kappa(i)\}$ and the corresponding set of all possible reproduction surfaces

[1]For clarity of presentation, we derive the prior probability in 1D. Generalization to 2D signals is presented in the Appendix.

$$\begin{aligned} -\log p(\hat{\mathbf{s}}^t|\mathbf{y}^t) &= \sum_{v=1}^{V}\left((\mathbf{y}_v^t-\mathbf{d}_v^t)^T\mathbf{Q}_v^t(\mathbf{y}_v^t-\mathbf{d}_v^t)+\gamma\sum_{n=1}^{N}\left[\mathbf{d}_v^t-\frac{\mathbf{D}^n\mathbf{d}_v^t+\mathbf{D}^{-n}\mathbf{d}_v^t}{2}\right]^T\mathbf{W}_v^t(n)\left[\mathbf{d}_v^t-\frac{\mathbf{D}^n\mathbf{d}_v^t+\mathbf{D}^{-n}\mathbf{d}_v^t}{2}\right]\right)+K(\mathbf{y}^t)\\ &\overset{\Delta}{=} e(\mathbf{d}_1^t,\ldots,\mathbf{d}_V^t|\mathbf{y}^t), \end{aligned} \tag{9}$$

$\hat{\mathcal{S}}=\{\beta(i)\}$. Thus we define $r(\hat{\mathbf{s}})=|\kappa(i)|$ if $\hat{\mathbf{s}}=\beta(i)$ for some index i, where $|\kappa|$ denotes the length of bit string κ.[2]

However, towards the goal of efficiently optimizing the encoder, we approximate $r(\hat{\mathbf{s}})$ based on a simple model of the MVC codec. In this model, a set of depth maps $\mathbf{d}_1^t,\ldots,\mathbf{d}_V^t$ is encoded in blocks using either motion compensated or disparity compensated prediction for each block. For each block, the prediction residual $\mathbf{x}$ is uniformly quantized to a bin of volume Δ and optimally entropy coded using a code of length $-\log p$ bits, where p is the bin probability. Assuming $\mathbf{x}$ is normally distributed according to $\mathbf{x}\sim N(0,\sigma^2\mathbf{I})$, the bin probability is

$$p\approx\frac{\Delta}{(2\pi\sigma^{2n})^{1/2}}e^{-\frac{1}{2\sigma^2}||\mathbf{x}||^2}, \tag{10}$$

and hence the number of bits used to code the residual is $-\log p\approx a||\mathbf{x}||^2+b$ for some constants a and b. Thus, a proxy for the number of bits needed to code the residual is simply the energy of the residual. Hence we are able to use the following proxies for the number of bits used to code a block in each coding mode.

1) Motion Compensation Proxy: If a block at position $\mathbf{p}$ is predicted from a previous frame of the same view via *motion compensation* (MC), we write the energy E_m as

$$\begin{aligned} E_m(\mathbf{d}_v^t(\mathbf{p}),\mathbf{v}_v^t(\mathbf{p})) &= ||\mathbf{d}_v^t(\mathbf{p})-\mathbf{d}_v^{t-1}(\mathbf{p}+\mathbf{v}_v^t(\mathbf{p}))||^2\\ &+\alpha_t\sum_{\mathbf{q}\in\mathcal{N}_\mathbf{p}}||\mathbf{v}_v^t(\mathbf{p})-\mathbf{v}_v^t(\mathbf{q})||^2, \end{aligned} \tag{11}$$

where $\mathbf{v}_v^t(\mathbf{p})$ is the *motion vector* (MV) for block $\mathbf{d}_v^t(\mathbf{p})$, and $\mathcal{N}_\mathbf{p}$ is a set of spatial neighboring blocks' positions causal to $\mathbf{p}$ (e.g., left, top, and top-right). In words, (11) is the sum of energies of two residuals: i) the motion prediction residual, and ii) the difference between MVs for the current block $\mathbf{p}$ and its causal neighboring blocks. α_t determines the relative scaling of the two energies in terms of bits.

2) Disparity Compensation Proxy: If a block at position $\mathbf{p}$ is predicted from a frame of a neighboring view of the same instant via *disparity compensation* (DC), we write the energy E_d, like we did for E_m in (11), as

$$\begin{aligned} E_d(\mathbf{d}_v^t(\mathbf{p}),\mathbf{u}_v^t(\mathbf{p})) &= ||\mathbf{d}_v^t(\mathbf{p})-\mathbf{d}_{v-1}^t(\mathbf{p}+\mathbf{u}_v^t(\mathbf{p}))||^2\\ &+\alpha_v\sum_{\mathbf{q}\in\mathcal{N}_\mathbf{p}}||\mathbf{u}_v^t(\mathbf{p})-\mathbf{u}_v^t(\mathbf{q})||^2, \end{aligned} \tag{12}$$

where $\mathbf{u}_v^t(\mathbf{p})$ is the *disparity vector* (DV) for block $\mathbf{d}_v^t(\mathbf{p})$.

[2]In general, a codec is defined by a composition of mappings $\mathcal{X}\overset{\alpha}{\to}\mathbb{N}\overset{\kappa}{\to}\{0,1\}^*\overset{\kappa^{-1}}{\to}\mathbb{N}\overset{\beta}{\to}\hat{\mathcal{X}}$ comprising an encoder $\alpha:\mathcal{X}\to\mathbb{N}$ mapping an input to an integer, a lossless (invertible) code $\kappa:\mathbb{N}\to\{0,1\}^*$ mapping an integer to a binary string, and a decoder $\beta:\mathbb{N}\to\hat{\mathcal{X}}$ mapping an integer to an output.

3) Combining Proxies for Rate Term: We now combine the two defined proxies into a single rate term. For a given block $\mathbf{p}$ with motion and disparity vectors $\mathbf{v}_v^t(\mathbf{p})$ and $\mathbf{u}_v^t(\mathbf{p})$, MVC selects the prediction mode (between MC and DC) with the smaller number of bits (assuming the same distortion). Furthermore, MVC selects the motion and disparity vectors to minimize the overall number of bits per frame. Thus, we estimate the number of bits needed to encode depth maps $\mathbf{d}_1^t,\ldots,\mathbf{d}_V^t$ as

$$\begin{aligned} &r(\mathbf{d}_1^t,\ldots,\mathbf{d}_V^t|\mathbf{v}_1^t,\ldots,\mathbf{v}_V^t,\mathbf{u}_1^t,\ldots,\mathbf{u}_V^t)\\ &=\mu\sum_{v=1}^{V}\sum_{\mathbf{p}\in\mathcal{B}}\min\left\{E_m(\mathbf{d}_v^t(\mathbf{p}),\mathbf{v}_v^t(\mathbf{p})),\ E_d(\mathbf{d}_v^t(\mathbf{p}),\mathbf{u}_v^t(\mathbf{p}))\right\}+\nu, \end{aligned} \tag{13}$$

$$r(\mathbf{d}_1^t,\ldots,\mathbf{d}_V^t)=\min_{\mathbf{v}_1^t,\ldots,\mathbf{v}_V^t,\mathbf{u}_1^t,\ldots,\mathbf{u}_V^t}r(\mathbf{d}_1^t,\ldots,\mathbf{d}_V^t|\mathbf{v}_1^t,\ldots,\mathbf{v}_V^t,\mathbf{u}_1^t,\ldots,\mathbf{u}_V^t) \tag{14}$$

where $\mathcal{B}$ is the set of coordinates for blocks in each $M\times N$ depth map, and μ and ν are constants. Finally, assuming that the codec would map the ideal depth maps for a reproduction $\hat{\mathbf{s}}^t$ to the reproduction $\hat{\mathbf{s}}^t$ itself, we define

$$r(\hat{\mathbf{s}}^t)=r(\mathbf{d}_1(\hat{\mathbf{s}}^t),\ldots,\mathbf{d}_V(\hat{\mathbf{s}}^t)). \tag{15}$$

C. Consistency Term

We need a consistency constraint $E_c(\mathbf{d}_1^t,\ldots,\mathbf{d}_V^r)<\eta$ to ensure that depth maps $\mathbf{d}_1^t,\ldots,\mathbf{d}_V^r$ are projections of a single 3D surface. Such a consistency constraint can be given in terms of the energy E_c of the differences between the depth maps when re-projected into other views,

$$E_c(\mathbf{d}_1^t,\ldots,\mathbf{d}_V^r)=\sum_{v=1}^{V}\sum_{j\neq v}\Delta\left(\mathbf{d}_v^t,\Phi_{j,v}(\mathbf{d}_j^t)\right) \tag{16}$$

where $\Phi_{j,v}(\mathbf{d})$ is a mapping function that maps pixels in $\mathbf{d}$ of view j to pixels in view v, and $\Delta(\mathbf{d}_1,\mathbf{d}_2)$ returns the average l_2-norm of the per-pixel difference between depth maps $\mathbf{d}_1$ and $\mathbf{d}_2$, computed using *only* corresponding pixels in $\mathbf{d}_1$ and $\mathbf{d}_2$ that are valid entries[3].

In practice, we compute the projected depth map $\tilde{\mathbf{d}}_v=\Phi_{j,v}(\mathbf{d}_j^t)$ as follows, assuming axis-aligned, rectified cameras. We first initialize $\tilde{\mathbf{d}}_v$ to be a matrix of invalid entries ∞. Because the mapping location must be an integer, a depth pixel $\mathbf{d}_j^t(x,y)$ at row x and column y is mapped to location $\tilde{\mathbf{d}}_v(x,y')$, where y' is computed as

$$y'=y+\text{round}\left(\frac{\varsigma}{\mathbf{d}_j^t(x,y)}\right), \tag{17}$$

[3]The projected map $\Phi_{j,v}(\mathbf{d}_j^t)$ may contain missing pixels due to occlusion, rounding, out-of-view problems, etc.

where ς is a scaling factor taking into account of the distance between neighboring cameras. Location (x, y') of projected map $\tilde{\mathbf{d}}_i$ is then updated as follows:

$$\tilde{\mathbf{d}}_v(x, y')) = \min\left\{\tilde{\mathbf{d}}_v(x, y'), \mathbf{d}_j^t(x, y)\right\}. \tag{18}$$

In other words, only the smaller of the new and previous values of $\tilde{\mathbf{d}}_v(x, y')$ (pixel closer to the camera) is kept as the visible pixel.

D. Optimization Problem

Given $\mathbf{d}_1^t = \mathbf{d}_1(\hat{\mathbf{s}}^t), \ldots, \mathbf{d}_V^t = \mathbf{d}_V(\hat{\mathbf{s}}^t)$, we have seen in (9) that $-\log p(\hat{\mathbf{s}}^t|\mathbf{y}^t)$ can be written as a function $e(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t|\mathbf{y}^t)$ and likewise we have seen in (15) that $r(\hat{\mathbf{s}}^t)$ can be written as a function $r(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t)$. Thus we can re-express our objective (2) as (19)-(23). In (20) we minimize over all surfaces, not just those in the reproduction codebook $\hat{\mathcal{S}}$. In (21) we convert the minimization over all surfaces to a minimization over all depth maps that are consistent with some surface. In (22) we loosen the consistency constraint by allowing the depth maps to be nearly consistent, which is important in practice due to roundoff errors. And in (23) we convert the constrained problem into an unconstrained problem using Lagrangian relaxation. We assume that the inequalities (20) and (22) are nearly equalities. Therefore, instead of solving (19) for the rate-constrained MAP surface $\hat{\mathbf{s}}^t$ (which is intractable), we solve (23) for the optimized depth maps $\mathbf{d}_1^*, \ldots, \mathbf{d}_V^*$ (which is relatively easy). Finally, we use the MVC codec to encode the optimized depth maps $\mathbf{d}_1^*, \ldots, \mathbf{d}_V^*$ into a single bit string, decode the bit string into quantized depth maps, and post-process the quantized depth maps into a single surface $\hat{\mathbf{s}}^t$.

Solving (23) for $\mathbf{d}_1^*, \ldots, \mathbf{d}_V^*$ involves a secondary minimization over motion and disparity vectors, as we can see in (24) by substituting (14) into (23), where our minimization objective is (25) for some $\lambda > 0$ and $\alpha_c > 0$. This minimization can be done iteratively, as described in Section VI-B.

V. Alternative Problem Formulation

In this section, we point out that it is also possible to perform *rate-constrained maximum likelihood estimation* by removing from (25) the smoothness term,

$$E_s(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t) = \gamma \sum_{v=1}^{V} \sum_{n=1}^{N} \left[\mathbf{d}_v^t - \frac{\mathbf{D}^n \mathbf{d}_v^t + \mathbf{D}^{-n} \mathbf{d}_v^t}{2}\right]^T \mathbf{W}_v^t(n) \left[\mathbf{d}_v^t - \frac{\mathbf{D}^n \mathbf{d}_v^t + \mathbf{D}^{-n} \mathbf{d}_v^t}{2}\right], \tag{26}$$

which corresponds to the negative logarithm of the prior probability, $-\log p(\mathbf{s}^t)$. This is the approach taken in our previous work [25]. Specifically, in [25], we defined *distortion* $d(\mathbf{y}^t, \mathbf{s}^t)$ as:

$$d(\mathbf{y}^t, \mathbf{s}^t) = -\log p(\mathbf{y}^t|\mathbf{s}^t) = \sum_{v=1}^{V} (\mathbf{y}_v^t - \mathbf{d}_v(\mathbf{s}^t))^T \mathbf{Q}_v^t (\mathbf{y}_v^t - \mathbf{d}_v(\mathbf{s}^t)), \tag{27}$$

and we defined *rate* $r_{ML}(\hat{\mathbf{s}}^t)$ in an analogous manner to $r(\hat{\mathbf{s}}^t)$ in (13)-(15). With these definitions, it is clear that the rate-distortion optimized encoder

$$\arg\min_{\hat{\mathbf{s}}^t \in \hat{\mathcal{S}}} d(\mathbf{y}^t, \hat{\mathbf{s}}^t) + \lambda_{ML} r_{ML}(\hat{\mathbf{s}}^t) \tag{28}$$

performs rate-constrained maximum likelihood estimation, i.e., finds $\hat{\mathbf{s}}^t \in \hat{\mathcal{S}} \subseteq \mathcal{S}$ that maximizes the likelihood $p(\mathbf{y}^t|\hat{\mathbf{s}}^t)$ subject to $r_{ML}(\hat{\mathbf{s}}^t) \leq \bar{R}$ for some $\bar{R}$. Moreover, the rate term $r_{ML}(\hat{\mathbf{s}}^t)$ determines a "prior" distribution over the codewords $\hat{\mathbf{s}}^t \in \hat{\mathcal{S}}$,

$$p(\hat{\mathbf{s}}^t) = 2^{-r_{ML}(\hat{\mathbf{s}}^t)}, \tag{29}$$

since $r_{ML}(\hat{\mathbf{s}}^t)$ is optimal (i.e., has expected value equal to the entropy) if and only if $r_{ML}(\hat{\mathbf{s}}^t) = -\log p(\hat{\mathbf{s}}^t)$. In general, $p(\hat{\mathbf{s}}^t)$ is not equal to the prior distribution over surfaces $p(\mathbf{s}^t)$, but is rather a discrete distribution equal to the marginal distribution resulting from the information theoretic problem,

$$\min_{p(\hat{\mathbf{s}}^t|\mathbf{y}^t)} I(\mathbf{Y}^t; \hat{\mathbf{S}}^t) \quad \text{s.t.} \quad d(\mathbf{Y}^t; \hat{\mathbf{S}}^t) \leq \bar{D}, \tag{30}$$

for some average distortion $\bar{D}$, assuming the encoder and decoder are jointly optimal, in the limit of large block size. However, as $\bar{D}$ gets small, $p(\hat{\mathbf{s}}^t)$ approaches $p(\mathbf{s}^t)$ in distribution, and hence under these conditions the rate-distortion optimized encoder (28), for $\lambda_{ML} = 1$, performs not only rate-constrained maximum likelihood estimation, but also unconstrained *maximum a posteriori* estimation. Indeed, when (26) is used as a component of the rate term $r_{ML}(\hat{\mathbf{s}}^t)$ (as it is in [25]), solving the rate-constrained ML problem (28) is essentially equivalent to solving the rate-constrained MAP problem (2) when $\lambda_{ML} = \lambda + 1$, and reduces to unconstrained maximum likelihood estimation when $\lambda_{ML} = 0$. This provides additional insight into the relationship between the rate-constrained (and unconstrained) ML and MAP problems.

VI. Optimization Algorithm

To arrive at a properly defined optimization problem, we first discuss in this section how we define diagonal weight matrix $\mathbf{W}_v^t(n)$ and precision matrix $\mathbf{Q}_v^t$ in (9). Then, we focus on the algorithm to solve our formulated optimization problem (25).

A. Define signal-adaptive weight and precision matrices

We use a *content-adaptive* approach to define $\mathbf{W}_v^t(n)$ and $\mathbf{Q}_v^t$. Diagonal weight matrix $\mathbf{W}_v^t(n)$ weights the respective samples in the neighborhood. In general, the smaller n is (closer neighbors), the larger the weight should be. Further, larger weights should be assigned to pixels on the same side of depth edges (same physical objects). This motivates us to find a kernel that can reveal both the spatial distance and the local image structure.

Bilateral filtering (BF) [26] defines weights between two pixels based on their spatial and photometric distances,

$$\begin{aligned}
\min_{\hat{\mathbf{s}}^t \in \hat{\mathcal{S}}} -\log p(\hat{\mathbf{s}}^t|\mathbf{y}^t) + \lambda r(\hat{\mathbf{s}}^t) &= \min_{\hat{\mathbf{s}}^t \in \hat{\mathcal{S}}} e(\mathbf{d}_1(\hat{\mathbf{s}}^t), \ldots, \mathbf{d}_V(\hat{\mathbf{s}}^t)|\mathbf{y}^t) + \lambda r(\mathbf{d}_1(\hat{\mathbf{s}}^t), \ldots, \mathbf{d}_V(\hat{\mathbf{s}}^t)) && (19)\\
&\geq \min_{\mathbf{s}^t \in \mathcal{S}} e(\mathbf{d}_1(\mathbf{s}^t), \ldots, \mathbf{d}_V(\mathbf{s}^t)|\mathbf{y}^t) + \lambda r(\mathbf{d}_1(\mathbf{s}^t), \ldots, \mathbf{d}_V(\mathbf{s}^t)) && (20)\\
&= \min_{\mathbf{d}_1^t,\ldots,\mathbf{d}_V^t} e(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t|\mathbf{y}^t) + \lambda r(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t) \quad \text{s.t.} \quad E_c(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t) = 0 && (21)\\
&\geq \min_{\mathbf{d}_1^t,\ldots,\mathbf{d}_V^t} e(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t|\mathbf{y}^t) + \lambda r(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t) \quad \text{s.t.} \quad E_c(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t) < \eta && (22)\\
&= \min_{\mathbf{d}_1^t,\ldots,\mathbf{d}_V^t} e(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t|\mathbf{y}^t) + \lambda r(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t) + \alpha_c(E_c(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t) - \eta) && (23)
\end{aligned}$$

$$\begin{aligned}
&\min_{\hat{\mathbf{s}}^t \in \hat{\mathcal{S}}} -\log p(\hat{\mathbf{s}}^t|\mathbf{y}^t) + \lambda r(\hat{\mathbf{s}}^t)\\
&= \min_{\mathbf{d}_1^t,\ldots,\mathbf{d}_V^t,\mathbf{v}_1^t,\ldots,\mathbf{v}_V^t,\mathbf{u}_1^t,\ldots,\mathbf{u}_V^t} e(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t|\mathbf{y}^t) + \lambda r(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t|\mathbf{v}_1^t, \ldots, \mathbf{v}_V^t, \mathbf{u}_1^t, \ldots, \mathbf{u}_V^t) + \alpha_c(E_c(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t) - \eta)\\
&= \min_{\mathbf{d}_1^t,\ldots,\mathbf{d}_V^t,\mathbf{v}_1^t,\ldots,\mathbf{v}_V^t,\mathbf{u}_1^t,\ldots,\mathbf{u}_V^t} J(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t, \mathbf{v}_1^t, \ldots, \mathbf{v}_V^t, \mathbf{u}_1^t, \ldots, \mathbf{u}_V^t) + const, \qquad (24)
\end{aligned}$$

$$\begin{aligned}
&J(\mathbf{d}_1^t, \ldots, \mathbf{d}_V^t, \mathbf{v}_1^t, \ldots, \mathbf{v}_V^t, \mathbf{u}_1^t, \ldots, \mathbf{u}_V^t)\\
&= \sum_{v=1}^{V} \left((\mathbf{y}_v^t - \mathbf{d}_v^t)^T \mathbf{Q}_v^t (\mathbf{y}_v^t - \mathbf{d}_v^t) + \gamma \sum_{n=1}^{N} \left[\mathbf{d}_v^t - \frac{\mathbf{D}^n \mathbf{d}_v^t + \mathbf{D}^{-n} \mathbf{d}_v^t}{2} \right]^T \mathbf{W}_v^t(n) \left[\mathbf{d}_v^t - \frac{\mathbf{D}^n \mathbf{d}_v^t + \mathbf{D}^{-n} \mathbf{d}_v^t}{2} \right] \right)\\
&+ \lambda \sum_{v=1}^{V} \sum_{\mathbf{p} \in \mathcal{B}} \min \left\{ E_m(\mathbf{d}_v^t(\mathbf{p}), \mathbf{v}_v^t(\mathbf{p})),\ E_d(\mathbf{d}_v^t(\mathbf{p}), \mathbf{u}_v^t(\mathbf{p})) \right\} + \alpha_c \sum_{v=1}^{V} \sum_{j \neq v} \Delta \left(\mathbf{d}_v^t, \Phi_{j,v}(\mathbf{d}_j^t) \right) \qquad (25)
\end{aligned}$$

but the definition of neighborhood (kernel) does not change with local structure. Thus, two near pixels with very different color intensities (large noise) will have almost no effect on each other due to a very small computed weight, even if they lie on the same side of a depth edge. Unlike BF, *steering kernel* [27] first identifies the local structure to define a locally adaptive kernel, using which a local weighted average is computed for denoising. As an example, Fig. 3(b) (from [27]) shows an example of the steering kernel, where the kernel shapes of pixels close to a detected edge become elongated ellipses to adapt to the local structure, as opposed to the fixed kernel in Fig. 3(a) for BF.

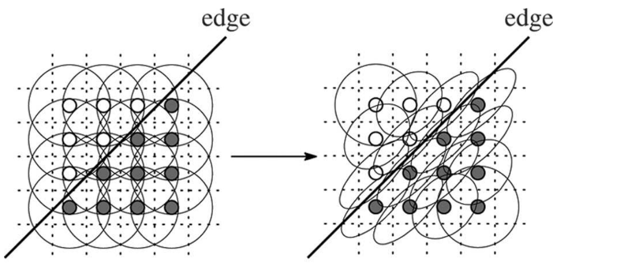


Fig. 3. Fixed versus adaptive kernels for image denoising: (a) fixed kernels employed by classical denoising schemes such as bilateral filters [26]; (b) adaptive kernels change according to local structures [27].

In particular, the steering kernel takes the form

$$K_i^{steer}(\mathbf{x}_i - \mathbf{x}) = \frac{\sqrt{det(\mathbf{C}_i)}}{2\pi h^2 \mu_i^2} \exp\left(-\frac{(\mathbf{x}_i - \mathbf{x})^T \mathbf{C}_i (\mathbf{x}_i - \mathbf{x})}{2h^2 \mu_i^2} \right) \qquad (31)$$

where $\mathbf{C}_i$ is the gradient covariance matrix based on differences in local values, $\mathbf{x}_i$ and $\mathbf{x}$ are the center pixel coordinate and neighbor pixel coordinate respectively, h is a scalar controlling the filtering strength, μ_i is the mean of the samples centered around $\mathbf{x}_i$. In (31), the influence of a neighboring pixel is tempered around local edges, with the appropriate choice of $\mathbf{C}_i$. To see the direct relation between the kernel shape and the covariance matrix, we decompose it into three components (equivalent to eigenvalue decomposition) as:

$$\begin{aligned}
\mathbf{C}_i &= \gamma_i \mathbf{R}_{\theta_i} \Gamma_i \mathbf{R}_{\theta_i}{}^T && (32)\\
\mathbf{R}_{\theta_i} &= \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} && (33)\\
\Gamma_i &= \begin{bmatrix} \sigma_i & 0 \\ 0 & \sigma_i^{-1} \end{bmatrix} && (34)
\end{aligned}$$

where $\mathbf{R}_{\theta_i}$ is the rotation matrix, and Γ_i is the elongation matrix. Note that the covariance is given by three parameters, σ_i, θ_i, and γ_i, which are elongation, rotation and scaling parameters, respectively. Essentially, a circle (classical kernel shape) becomes an ellipse after the elongation matrix, then it rotates to the principle edge after the rotation matrix, and finally the suitable scale is determined by the scaling parameter.

Since the local edge structure is related to the gradient covariance (or equivalently, the locally dominant orientation), the three parameters σ_i, θ_i, and γ_i can therefore be determined from the gradient covariance. Following [27] which in addition has a noise tolerance property, the dominant orientation of the local gradient field is the singular vector corresponding to the smallest (nonzero) value of the local gradient matrix G_i:

$$G_i = \begin{bmatrix} \vdots & \vdots \\ g_x(j) & g_y(j) \\ \vdots & \vdots \end{bmatrix} = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^T, \qquad j \in w_i \tag{35}$$

where $g_x(\cdot)$ and $g_y(\cdot)$ are the first derivatives along x and y directions, $\mathbf{U}_i$, $\mathbf{S}_i$, $\mathbf{V}_i$ are matrices after *Singular Value Decomposition* (SVD), and w_i is the local analysis window. The three parameters in $\mathbf{C}_i$ can thus be defined as follows:

$$\theta_i = \arctan\left(\frac{\mathbf{V}_i(1,2)}{\mathbf{V}_i(2,2)}\right) \tag{36}$$

$$\sigma_i = \frac{\mathbf{S}_i(1,1)+\epsilon'}{\mathbf{S}_i(2,2)+\epsilon'} \tag{37}$$

$$\gamma_i = \sqrt{\frac{\mathbf{S}_i(1,1)\mathbf{S}_i(2,2)+\epsilon''}{N}} \tag{38}$$

where $\mathbf{V}_i(a,b)$ and $\mathbf{S}_i(a,b)$ are the elements in the a^{th} row and b^{th} column of matrices $\mathbf{V}_i$ and $\mathbf{S}_i$, respectively. ϵ' and ϵ'' are two constants for stability consideration. N is the number of samples in the local analysis window.

1) Estimate the weighting matrix: After the kernel for one pixel is determined, the filtering weights of its neighboring pixels are determined straightforwardly using (31). In this paper[4], $\mathbf{W}(n)$ is a diagonal matrix which captures the weights for all pair of pixels with spatial distances $\mathbf{x}_i - \mathbf{x}$ equal to n.

$$\mathbf{W}(n)(i,i) = K_i^{steer}(n) \tag{39}$$

In other words, for pixel i, the weight of its neighboring pixel, with the spatial distance being n, is defined in the $(i,i)^{th}$ entry of the weighting matrix $\mathbf{W}(n)$, and also in its steering kernel $K_i^{steer}(n)$.

2) Estimate the precision matrix: Precision matrix can be computed as the inverse of the covariance matrix $\mathbf{\Sigma}$. However, the matrix inverse operation is very computation-intensive and may not be numerically stable if insufficient number of samples are used. Therefore, instead of estimating $\mathbf{\Sigma}$ and then computing $\mathbf{Q} = \mathbf{\Sigma}^{-1}$, we approximate the precision matrix directly as follows. For the sake of simplicity, we assume that the precision matrix only have diagonal entries together with sparse non-zero off-diagonal entries. In [28], the relationship between the entries of $\mathbf{Q}$ and $\mathbf{\Sigma}$ is given by:

$$var(i) = \mathbf{\Sigma}(i,i) \approx \frac{1}{\mathbf{Q}(i,i)} \tag{40}$$

$$corr(i,j) = \frac{\mathbf{\Sigma}(i,j)}{\sqrt{\mathbf{\Sigma}(i,i)\mathbf{\Sigma}(j,j)}} = -\frac{\mathbf{Q}(i,j)}{\sqrt{\mathbf{Q}(i,i)\mathbf{Q}(j,j)}} \tag{41}$$

Rearranging (40) and (41), entries in the precision matrix can be approximated as:

$$\mathbf{Q}(i,i) \approx \frac{1}{\mathbf{\Sigma}(i,i)} \tag{42}$$

$$\mathbf{Q}(i,j) \approx -\frac{\mathbf{\Sigma}(i,j)}{\mathbf{\Sigma}(i,i)\mathbf{\Sigma}(j,j)} \tag{43}$$

In other words, the diagonal entries are the inverse of the variance of the samples, and the off-diagonal entries are scaled sample covariances by the diagonal entries.

Next, we adaptively collect samples to estimate the sample covariance matrix using the previously introduced steering kernel. Assuming the noises only have correlations on the same side of the edge, for each interested pixel, we collect neighboring samples on the same side of a possibly detected edge for averaging. Mathematically, a sample with the spatial distance to the center pixel being n is collected when $\frac{K_i^{steer}(n)}{K_i^{steer}(0)} \geq 0.5$. After collecting all the causal samples, the sample covariance can be computed in a general way as:

$$\mathbf{\Sigma}(i,j) = \mathbf{E}\Big((X_i - \mathbf{E}(X_i))(X_j - \mathbf{E}(X_j))\Big) \tag{44}$$

B. Optimization in Details

We now describe our algorithm to solve our formulated optimization problem (25). We first note that the inter-frame predictors, MVs $\mathbf{v}_v^t$ and DVs $\mathbf{u}_v^t$, and depth maps $\mathbf{d}_v^t$ are inter-dependent. To resolve the inter-dependency, we alternately optimize either depth $\mathbf{d}_v^t$ or predictors $\mathbf{v}_v^t$ and $\mathbf{u}_v^t$ at a time, until convergence. Specifically, given $\mathbf{d}_v^t$, we optimize $\mathbf{v}_v^t$ and $\mathbf{u}_v^t$ via block search to minimize our objective function. Then, given $\mathbf{v}_v^t$ and $\mathbf{u}_v^t$, we optimize $\mathbf{d}_v^t$ to minimize the same objective.

Optimizing depth maps $\mathbf{d}_v^t$ for fixed $\mathbf{v}_v^t$ and $\mathbf{u}_v^t$ is still difficult, due to the non-convex mapping function $\phi_{i,k}(\cdot)$ in the consistency term (16). We thus propose an alternating two-step procedure as follows. The two steps are: i) align edges in depth maps of consecutive views to match *scene structure* across views; and ii) smooth surfaces within depth edges to match *scene texture* across views. We optimize one view a time while fixing the other views. Note that the variables in the mapping function $\phi_{i,k}(\cdot)$ become fixed when optimizing one view only, simplifying our optimization. An overview of this algorithm is described below. (Description of block search, a common procedure in image / video processing [29]–[31], is omitted.)

[4]For clarity of presentation, we derive the equations in 1D. Generalization to 2D signals is presented in the Appendix.

Algorithm 1 Overview of the Optimization Algorithm

```
1: Initialize $\alpha_c = 0$, $\mathbf{d}_i^t = \mathbf{y}_i^t$.
2: Estimate the weight matrices $\mathbf{W}(\mathbf{n})$ from $\mathbf{d}_i^t$.
3: Estimate the precision matrices $\mathbf{Q}_v^t$'s from noisy observations $\mathbf{y}_i^t$.
4: repeat
5:     repeat
6:         Given $\mathbf{d}_i^t$, find optimal $\mathbf{v}_i^t$ and $\mathbf{u}_i^t$ by motion search.
7:         Given $\mathbf{v}_i^t$ and $\mathbf{u}_i^t$, match scene structure by edge realignment. {Step A}
8:         Given $\mathbf{v}_i^t$ and $\mathbf{u}_i^t$, match scene texture by surface smoothing. {Step B}
9:     until MV and DV converge.
10:    Increase $\alpha_c$.
11: until $\alpha_c$ sufficiently large.
```

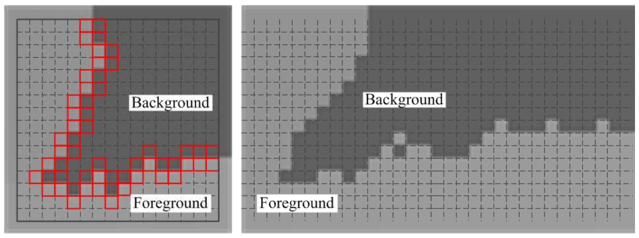


Fig. 4. Inconsistencies of the distinct edges across views. Left: block in current view. Right: corresponding blocks in reference view.

1) Step A: Match scene structure by edge realignment: Typical depth maps are piecewise smooth. Hence large inconsistency across views usually occurs when a pixel in a nearer depth region (foreground) in one view is mapped to a pixel in a farther depth region (background) in a neighboring view (or vice versa), resulting in a large increase in the consistency term (16). Fig. 4 illustrates example blocks with foreground and background regions and distinct edges between them.

To correct for these large consistency errors, we attempt to align the boundaries of regions across views; i.e., we match the scene structure across views. Specifically, we first detect depth edges in a block using a simple thresholding method: we declare an edge between two neighboring pixels if the depth values between them is larger than a threshold ϵ.

Pixels on either side of a declared edge are labeled *candidate pixels*. At each iteration, we test the reassignment of opposite depth values at each candidate pixel (from foreground to background or vice versa), and note the potential decrease in objective (25). The candidate pixel with the largest decrease in objective is chosen for depth value reassignment. The depth value reassignment induces a change in the set of detected edges and candidate pixels, so both are updated correspondingly. To make the reassignment operation robust to noise, the depth value of a candidate pixel will be reassigned only if the resulting consistency E_c decreases by a significant amount. We repeat this process until there are no more depth value reassignment of candidate pixels that can induce a further decrease in objective value.

Note that the above edge realignment procedure is performed on a *target* depth map in a single view given a set of *reference* depth maps of a neighboring view are fixed and used for computation of (16). The complete algorithm is shown below.

Algorithm 2 Step A: edge realignment

```
1: for $i = 1$ to $V$ do
2:     Detect edges and identify candidate pixels.
3:     repeat
4:         repeat
5:             Test opposite depth assignment on candidates.
6:             Pick winner, update edge and candidate pixels.
7:         until No objective-decreasing candidates.
8:     until All blocks are processed.
9: end for
```

2) Step B: Match scene texture by surface smoothing: In this step, we match the interior regions (texture) given the depth edges (structure) of the neighboring views are now aligned. To minimize the objective function (25), we employ the Jacobi algorithm (also known as *diagonal normalized steepest descent* (DNSD) [24]). Generally, to minimize cost $\varepsilon(\mathbf{d})$, the locally optimal $\mathbf{d}$ can be computed in a single iteration of the Jacobi algorithm given initialized $\mathbf{d_0}$:

$$\mathbf{d} = \mathbf{d_0} - \mathbf{M}(\mathbf{d_0})\frac{\partial \varepsilon(\mathbf{d})}{\partial \mathbf{d}}|_{\mathbf{d}=\mathbf{d_0}} \tag{45}$$

where $\mathbf{M}(\mathbf{d_0})$ is a diagonal matrix with its diagonal entries being the locally adaptive step-sizes. $\mathbf{M}(\mathbf{d_0})$ is defined as the inverse of the main diagonal of the Hessian matrix (i.e., the second derivative of the cost function):

$$\mathbf{M}(\mathbf{d_0}) = [\xi \mathbf{I} + diag\{\mathrm{H}(\mathbf{d_0})\}]^{-1} \tag{46}$$

where $\xi\mathbf{I}$ is a scaled identity matrix to ensure stability.

We first write the partial derivative of objective (25) with respect to depth map $\mathbf{d}_i^t$ as:

$$\frac{\partial}{\partial \mathbf{d}_i^t} = 2\mathbf{Q}(\mathbf{d}_i^t - \mathbf{y}_i^t) + \gamma \sum_{n=1}^{N} (\mathbf{I} - \frac{\mathbf{D^n} + \mathbf{D^{-n}}}{2})\mathbf{W}(n)(\mathbf{I} - \frac{\mathbf{D^n} + \mathbf{D^{-n}}}{2})\mathbf{d}_i^t$$
$$+ 2\alpha_c \sum_{j \in N(i)} \left(\mathbf{d}_i^t - \phi_{j,i}(\mathbf{d}_j^t)\right) + 2\lambda\alpha_p \begin{cases} \mathbf{d}_i^t - \mathbf{d}_i^{t-1}(\mathbf{v}_i^t) & \text{motion mode} \\ \mathbf{d}_i^t - \mathbf{d}_{i-1}^t(\mathbf{u}_i^t) & \text{disparity mode} \end{cases} \tag{47}$$

For notation simplicity, we next define a temporary term $A = \sum_{n=1}^{N}(\mathbf{I} - \frac{\mathbf{D^n}+\mathbf{D^{-n}}}{2})\mathbf{W}(n)(\mathbf{I} - \frac{\mathbf{D^n}+\mathbf{D^{-n}}}{2})\mathbf{d}_i^t$. The k^{th} sample of A can be rewritten as:

$$A(k) = 2\sum_{n=1}^{N} \mathbf{W}(n)(k,k)\left(\mathbf{d}_i^t(k) - \frac{\mathbf{d}_i^t(k-n) + \mathbf{d}_i^t(k+n)}{2}\right) \tag{48}$$

By taken one more partial derivative of (47) with (48) inserted, the second derivative (i.e., Hessian matrix) of the objective (25) is:

$$H = \frac{\partial^2}{\partial(\mathbf{d}_i^t)^2} = 2\mathbf{Q} + \gamma \sum_{n=-N}^{N} \mathbf{W}(n) + 2V\alpha_c\mathbf{I} + 2\lambda\alpha_p\mathbf{I} \tag{49}$$

Combining (46) and (49), $(k,k)^{th}$ entry of $\mathbf{M}$ is:

$$\mathbf{M}(k,k) = \frac{1}{\xi + 2\mathbf{Q}(k,k) + \gamma \sum_{n=-N}^{N} \mathbf{W}(n)(k,k) + 2V\alpha_c + 2\lambda\alpha_p} \tag{50}$$

If we apply a single iteration of the Jacobi algorithm with initialized $\mathbf{d_0}$, for the k^{th} sample, we can derive $\mathbf{d}_i^t(k)$ as a time varying convolution of the form:

$$\mathbf{d}_i^t(k) = \sum_{l=-N}^{N} f(k,l)\mathbf{d_0}(l) + \Lambda(k) \tag{51}$$

where the filter's coefficients $f(k,l)$ is given by

$$f(k,l) = \begin{cases} \mathbf{M}(k,k)\left(\xi + \gamma\mathbf{W}(0)(k,k)\right), & l = 0 \\ \mathbf{M}(k,k)\left(\gamma\mathbf{W}(l)(k,k) - 2\mathbf{Q}(k,k+l)\right), & l \neq 0 \end{cases} \tag{52}$$

Assuming the motion mode is selected, the last term $\Lambda(k)$ is expressed as

$$\begin{aligned}\Lambda(k) = \mathbf{M}(k,k)\Big(&\sum 2\mathbf{Q}(k,k+l)\mathbf{y}_i^t(k+l) \\ &+ 2\alpha_c \sum_{j\in\mathcal{N}(i)} \phi_{j,i}(\mathbf{d}_j^t)(k) + 2\lambda\alpha_p\mathbf{d}_i^{t-1}(k+\mathbf{v}_i^t(k))\Big)\end{aligned} \tag{53}$$

Similar form of solution can be derived if the disparity mode is selected.

Recall that the non-convex problem (due to non-convex mapping function) relaxes to a convex one by optimizing one frame at a time. The Jacobi algorithm described above minimizes our objective in (25) with neighboring views fixed in each iteration. We observe that when λ is large, view prediction $\mathbf{d}_{i-1}^t(k+\mathbf{u}_i^t(k))$ have strong influence on the current view $\mathbf{d}_i^t(k)$. In such case, if there are large errors in $\mathbf{d}_{i-1}^t(k+\mathbf{u}_i^t(k))$ from the solution in (51) and (53), they are easily propagated to the current view. This causes the algorithm to be trapped in a local minimum.

Intuitively, when λ is large, we know that depth maps with little textual details have smaller objective values. Inspired by *simulated annealing* [32], we attempt to escape from local minima by smoothing out textural details in the prediction using a simple box filter $\mathbf{h}_l$ of size $l \times l$:

$$\mathbf{h}_l = \frac{1}{l^2}\begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \vdots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}_{l\times l}$$

where the smoothed prediction is the convolution $\mathbf{d}_{i-1}^t(k+\mathbf{u}_i^t(k)) * \mathbf{h}_l$. By attempting a spatial filter in one of the search paths in simulated annealing, a prominent local error in a previous view will be averaged out, resulting in smaller prediction error energy and thus a smaller rate term. Among $l = \{3,5,7\}$ and no smoothing, we select the one that minimizes our objective. This simple procedure provides opportunities for our algorithm to escape from a local minimum when λ is large.

VII. Experimentation

A. Experimental Setup

To test the performance of our proposed algorithm, we used texture and depth maps from two 1024×768 MPEG FTV multiview test sequences, `Lovebird1` at camera-captured views $4,6,8$ and `Balloons` at views $1,2,3$, and one synthetic sequence `Dude` added with Gaussian noise at views $1,2,3$. The first 15 frames of each sequence were used in our experiments.

The test sequences were pre-processed using one of three methods before being compressed with a MVC codec [6]. In the first method, `Averaged`, we first applied an average procedure to the raw acquired depth maps $\mathbf{y}$, similar to one in [17], which projects all views to the center view, averages the projected depth values for each pixel, and then re-projects the center view back to the other views. In the second method `Optimized`, our proposed joint denoising / compression algorithm was used to produce a surface $\hat{\mathbf{s}}_\lambda$ that minimized (25) for a given value of λ. The surface $\hat{\mathbf{s}}_\lambda$ was projected onto the V views, and the resulting depth maps were fed into the MVC codec. In the third method, `MAP-solution`, our optimization algorithm was executed for $\lambda = 0$ to produce a MAP surface $\hat{\mathbf{s}}_0$. The surface $\hat{\mathbf{s}}_0$ was projected onto the V views and the resulting depth maps were fed into the codec. Because we set $\lambda = 0$, the most likely surface was sought with no consideration for the representation size, hence `MAP-solution` represents the separate denoising / compression approach.

In our implementation, we set the range of λ to be $\lambda \in \{0, 10^0, 10^1, 10^2, 10^3, 10^4, 10^5\}$, and the range of quantization parameter (QP) in MVC to be QP $\in \{15,20,25,27,30,32,35,37,40,42,45,50\}$. We will discuss the selection of λ and QP pairs shortly.

After encoding and decoding the set of depth maps using the MVC codec, the decoded depth maps may no longer be consistent across views. To ensure inter-view consistency at the decoder, we applied the averaging procedure again, similar to one in [17]. These post-processed depth maps represent the decoded surface. Note that if a lossless codec is used instead, then there is no need to do any post-processing.

Next, we will first discuss the selection of λ and QP pairs. Then we evaluate the quality of the decoded surface using two metrics: our *observation-surface distortion* metric for the depth maps and the *PSNR-bitrate* metric for the intermediate virtual view synthesized using the decoded depth maps.

B. Selection of λ and QP pairs

We now discuss how to select pairs of λ and QP for optimal operating points given multiview depth video is coded using a lossy codec like MVC. The bitrate of a lossy codec like MVC for coding of multiview video is determined by the chosen QP value. Large QP generally leads to high compression ratio but also severe coding artifacts. In our formulation (25), however, for any value of λ we seek a compactly represented surface $\mathbf{s}$ assuming $r(\mathbf{s})$ is the minimum rate that can sufficiently described surface $\mathbf{s}$ (without loss) to the decoder. In other words, we do not account for lossy compression of the codec itself after a compactly represented surface $\mathbf{s}$ is found. One straight-forward way to find the optimal pairs of λ and QP is to examine the performance (rate and distortion) of all pairs, then trace out the convex hull as the operational rate-distortion curve. While optimal, this procedure is computationally expensive, and may not be suitable for real-time applications.

For faster computation, we propose the following procedure. We search for the coarsest QP that still enables a sufficiently good surface reconstruction at decoder, for a given chosen surface $\hat{\mathbf{s}}_\lambda$, i.e., one that suffers no more than *just-noticeable-difference* (JND) [33] as compared to the original computed surface. To find such QP, for a given λ we perform binary search for the coarsest QP that can produce JND. We plot the optimal pairs of λ and QP in Fig. 5. We can observe from the figure that for large λ (low rate), a large QP (aggressive compression) can fulfill JND requirement, which agrees with our intuition.

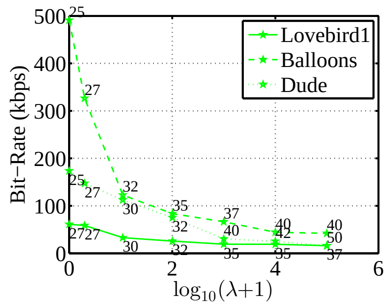


Fig. 5. Bitrate for the test sequences with the optimal combination of QP in the lossy codec and λ in our formulation.

C. Performance of Depth Map Reconstruction

The first metric is our observation-surface distortion measure $-\log p(\hat{\mathbf{s}}^t_{QP}|\mathbf{y}^t)$ defined in (9), where $\hat{\mathbf{s}}^t_{QP}$ is the MVC encoded surface at a certain QP value.

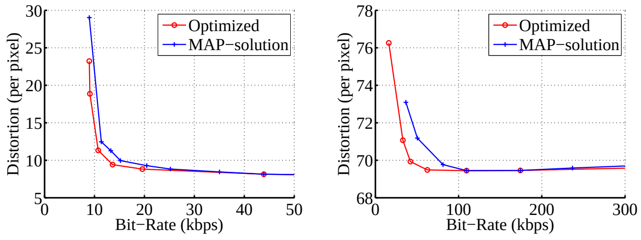


Fig. 6. Distortion-rate curve of the generated surfaces using the surface-observation distortion metric for `Lovebird1` and `Balloons`.

Figure 6 and Figure 7 show distortion-rate curves of `Optimized` and `MAP-solution` for the `Lovebird1`, `Balloons` and `Dude` sequences, respectively. Each curve for `MAP-solution` was generated by varying QP in the MVC codec, while each RD curve for `Optimized` is the lower convex hull of all RD pairs generated by varying both QP and λ. We can observe from Figure 6 and Figure 7 that at lower bit-rate, `Optimized` outperformed `MAP-solution`. We observe also that at higher bit-rate, `Optimized` converged to `MAP-solution`, which is what we expect as the rate constraint is loosened. Hence the experimental results demonstrate that the joint denoising / compression approach (`Optimized`) indeed outperforms the conventional separate denoising / compression approach (`MAP-solution`) when the rate constraint is tight.

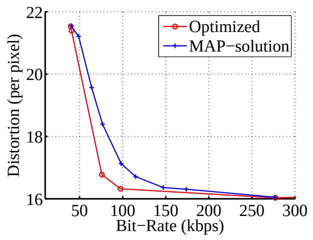


Fig. 7. Distortion-rate curve of the generated surfaces using the surface-observation distortion metric for `Dude`.

Figures 8 shows the raw acquired depth maps and MAP depth maps from different views for `Lovebird1`, `Balloons` and `Dude`. Compared to the raw acquired depth maps, the MAP depth maps show a visually significant improvement in inter-view consistency as pointed by arrows in `Lovebird1` and `Balloons`, as well as noise reduction in `Dude`.

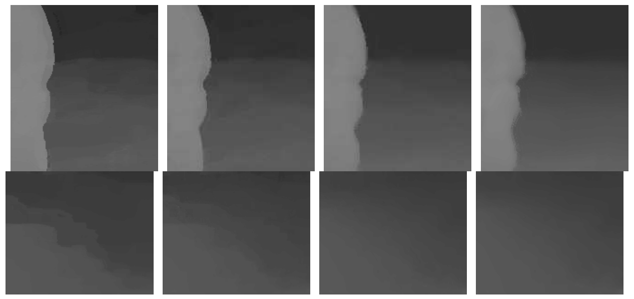


Fig. 9. Cropped regions of Optimized depth map with $\lambda = 0, 1, 10^2, 10^4$, shown from left to right respectively. Smoothed details and smeared edges are observed for large λ.

Figure 9 demonstrates the typical example of optimized depth maps in the center view for $\lambda = 0, 1, 10^2, 10^4$. As λ increased, the rate term became a heavier penalty, resulting in a larger distortion. We observe that the depth map for $\lambda = 100$ has less details inside the foreground and background regions, and smeared edges between foreground and background. Without sharp edges, a 3D surface becomes easier to code and therefore a smaller rate term $r(\mathbf{s})$.

D. Performance of Depth-Image-Based Rendering

The second metric is the PSNR of an intermediate virtual view (between neighboring camera views) synthesized from decoded texture and depth maps, via DIBR [2]. For this metric, the ground truth is taken to be the original texture map provided by the sequence provider at the virtual viewpoint.

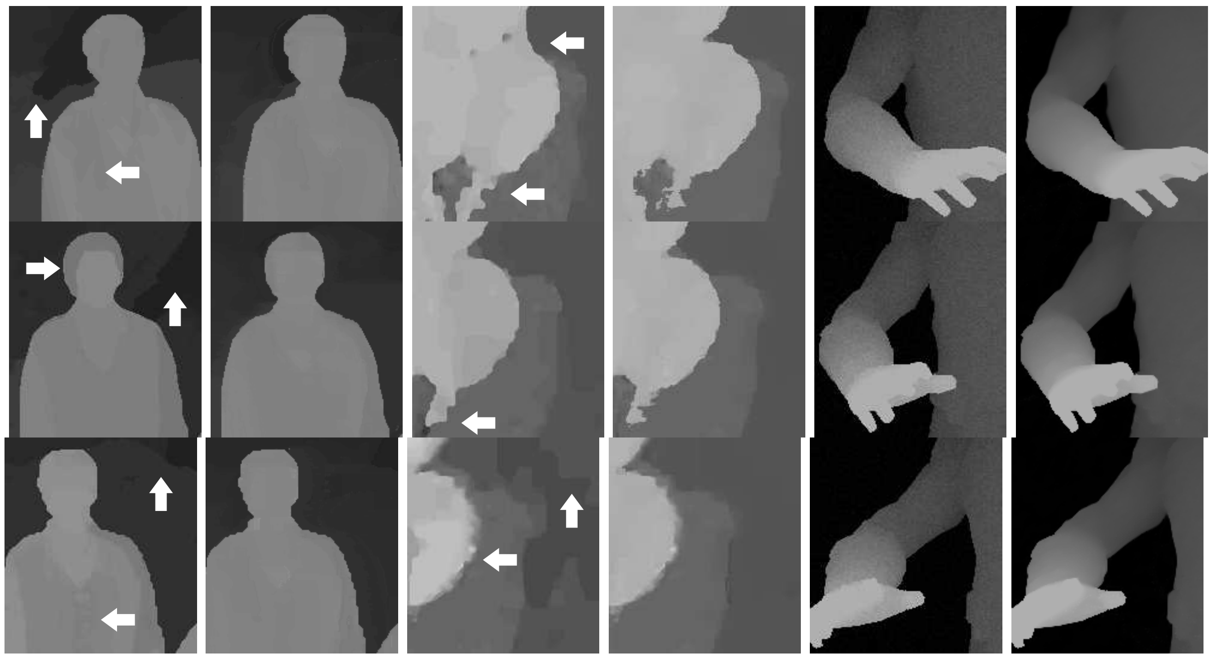


Fig. 8. Comparing the raw acquired depth maps with MAP depth maps for `Lovebird1`, `Balloons` and `Dude` at three different views (in order to see differences clearly, all pixel values are doubled). From left to right: Raw acquired depth maps of `Lovebird1` at views 4,6,8; MAP depth maps of `Lovebird1` at views 4,6,8; Raw acquired depth maps of `Balloons` at views 1,3,5; MAP depth maps of `Balloons` at views 1,3,5; Raw acquired depth maps of `Dude` at views 1,2,3; and MAP depth maps of `Dude` at views 1,2,3.

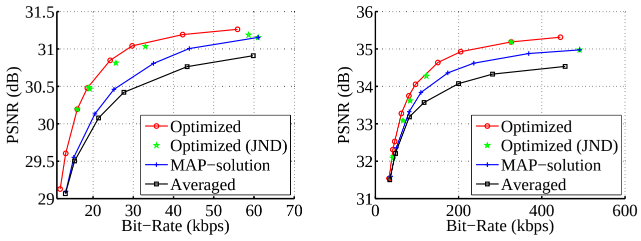


Fig. 10. Synthesized view PSNR at decoder vs. coding rate for `Lovebird1` and `Balloons`.

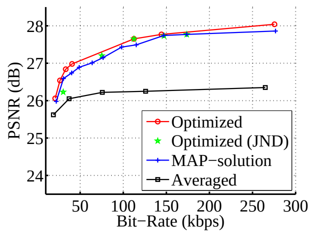


Fig. 11. Synthesized view PSNR at decoder vs. coding rate for `Dude`.

Fig. 10 and 11 show the PSNR-rate curve for two natural sequences `Lovebird1` and `Balloons`, and the synthetic sequence `Dude` respectively. The PSNR-rate curves for `Averaged` and `MAP-solution` were generated by varying the MVC quantization parameters (QPs), while the RD curve for `Optimized` is the convex hull of all RD pairs generated by varying both QP and λ. The RD points for `Optimized (JND)` are generated using the JND QP selection procedure described in Section VII-B. One can observe that `Optimized (JND)` is very close to the RD-curve `Optimized`. This verifies that, when using MVC to compress the optimized surfaces, one can practically decide the suitable QPs without trying all λ / QP combinations. At lower bitrate, using the Bjontegaard metric to compute rate reduction, for `Lovebird1`, the average rate for `Optimized` is 76.90% of `MAP-solution` and of 67.94% `Averaged`, while achieving the same PSNR. The respective percentages of coding rate for `Balloons` are 78.52% and 61.86%, and for `Dude` are 68.69% and 62.34%. As rate increased, 3D surfaces computed by `Optimized` approached those computed by `MAP-solution`, thus achieving the same PNSR-rate performance, as expected.

Cropped images at virtual views of the three sequences are shown in Fig. 12. Improvements indicated by arrows can be clearly observed.

VIII. Conclusion

Given noise-corrupted depth observations from multiple viewpoints, in this paper we propose to construct a rate-constrained 3D surface of a dynamic scene subject to a representation size constraint. Unlike previous work that finds the most likely 3D surface given noisy observations regardless of representation size, our identified 3D surface optimally trades off the posterior probability with representation size. We propose an iterative algorithm that alternately optimizes the scene structure (depth edges) and the scene texture (depth surfaces) until convergence. Experimental results show that using projections of our rate-constrained 3D reconstruction to multiple depth maps for multiview depth video coding can reduce coding rate of depth maps by up to 32% compared to unconstrained estimated surfaces for the same quality of synthesized virtual views at the decoder.

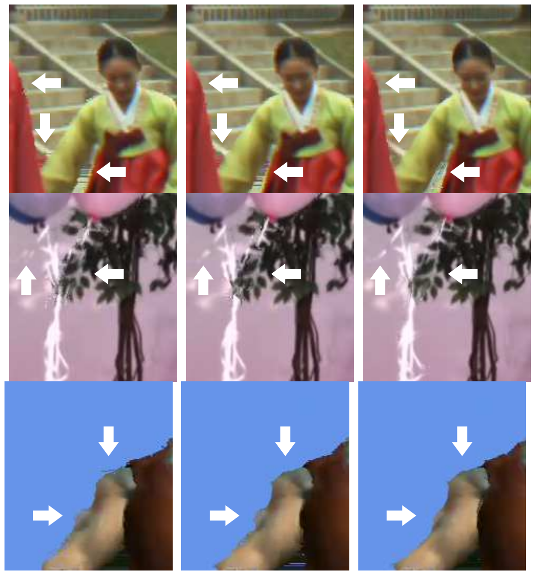


Fig. 12. Top Row (`Lovebird1`): synthesized virtual view 5 using texture and depth maps at view 4 and 6. Depth maps are of 48kbps: `Averaged` (left), `MAP-solution` (center), `Optimized` (right). Center Row (`Balloons`): synthesized virtual view 4 using texture depth maps at view 3 and 5. Depth maps are of 120kbps: `Averaged` (left), `MAP-solution` (center), `Optimized` (right). Bottom Row (`Dude`): synthesized virtual view 2 using texture depth maps at view 1 and 3. Depth maps are of 50kbps: `Averaged` (left), `MAP-solution` (center), `Optimized` (right).

Appendix

A. Generalization to 2D signals

In the appendix, we generalize the prior probability defined in Section IV for a 2D signal. Without loss of generality, assume $\mathbf{d}$ is the column-ordered vector of the 2D signal, the shift operator $\mathbf{D}$ and the penalty function $\mathbf{W}(\mathbf{n})$ need to be updated accordingly. Note that as the signal goes from 1D to 2D, the shift direction $\mathbf{n}$ also goes from 1D to 2D, which is why we use the boldface here. For example, $\mathbf{d}$ is a column ordered vector of a 3×3 2D signal, then the one-sample right-down circular shift operator on this input vector is defined as:

$$\mathbf{D} = \begin{bmatrix} 0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1&0 \\ 0&0&1&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \end{bmatrix} \tag{54}$$

The penalty function $\mathbf{W}(\mathbf{n})$ keeps as an diagonal matrix, but the input to this function changes from 1D to 2D. As it is defined by the steering kernel (31), the 2D input signals apply directly without any modification.

References

[1] S. B. Gokturk, H. Yalcin, and C. Bamji, "A time-of-flight depth sensor system description, issues and solutions," in *CVPR Workshop*, Washington, DC, June 2004.

[2] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila, "View synthesis techniques for 3D video," in *App. of Digital Image Processin XXXII, Proc. of the SPIE*, vol. 7443 (2009), 2009, pp. 74 430T–74 430T–11.

[3] Z. Zhang and O. Faugeras, "A 3D world model builder with a mobile robot," in *International Journal of Robotics Research*, vol. 11, no.4, August 1992, pp. 269–285.

[4] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *ACM SIGGRAPH*, New Orleans, LA, August 1996.

[5] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fizgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *IEEE Int. Symp. on Mixed and Augmented Reality*, Basel, Switzerland, October 2011.

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

[7] B. Huhle, T. Schairer, P. Jenke, and W. Straber, "Robust non-local denoising of colored depth data," in *IEEE CVPR Workshop on Time of Flight Camera based Computer Vision*, Anchorage, AK, June 2008.

[8] M. Tallón, S. D. Babacan, J. Mateos, M. Do, R. Molina, and A. Katsaggelos, "Upsampling and denoising of depth maps via joint-segmentation," in *20th European Signal Processing Conference (EUSIPCO 2012)*, Bucharest, Romania, August 2012.

[9] D. Min, J. Lu, and M. Do, "Depth video enhancement based on weighted mode filtering," in *IEEE Transactions on Image Processing*, vol. 21, no.3, March 2012.

[10] S.-W. Jung, "Enhancement of image and depth map using adaptive joint trilateral filter," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no.2, February 2013.

[11] L. Chen, H. Lin, and S. Li, "Depth image enhancement for kinect using region growing and bilateral filter," in *IEEE International Conference on Pattern Recognition*, Tsukuba, Japan, November 2012.

[12] J. Shen and S. c. S. Cheung, "Layer depth denoising and completion for structured-light RGB-D cameras," in *IEEE International Conference on CVPR*, Portland, OR, June 2013.

[13] C. Chen, J. Cai, J. Zheng, T.-J. Cham, and G. Shi, "A color-guided, region-adaptive and depth-selective unified framework for kinect depth recovery," in *IEEE International Workshop on Multimedia Signal Processing*, Pula, Italy, October 2013.

[14] W. Hu, X. Li, G. Cheung, and O. Au, "Depth map denoising using graph-based transform and group sparsity," in *IEEE International Workshop on Multimedia Signal Processing*, Pula, Italy, October 2013.

[15] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no.7, July 2003, pp. 560–576.

[16] H. Helgason, H. Li, and M. Flierl, "Multiscale framework for adaptive and robust enhancement of depth in multi-view imagery," in *IEEE ICIP*, Orlando, FL, October 2012.

[17] R. Li, D. Rusanovskyy, M. Hannuksela, and H. Li, "Joint view filtering for multiview depth map sequences," in *IEEE ICIP*, Orlando, FL, October 2012.

[18] T. Fujii, K. Mori, K. Takeda, K. Mase, M. Tanimoto, and Y. Suenaga, "Multipoint measuring system for video and sound—100 camera and microphone system," in *IEEE International Conference on Multimedia and Expo*, Toronto, Canada, July 2006.

[19] J. Stam, "Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values," in *SIGGRAPH*, September 1988, pp. 395–404.

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

[21] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no.1, January 1989, pp. 31–42.

[22] Y. Gao, G. Cheung, T. Maugey, P. Frossard, and J. Liang, "3D geometry representation using multiview coding of image tiles," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Florence, Italy, May 2014.

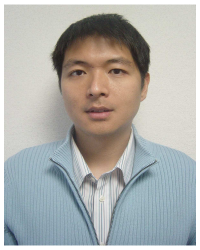
[23] Q. Cai, D. Gallup, C. Zhang, and Z. Zhang, "3D deformable face tracking with commodity depth cameras," in *ECCV*, Crete, Greece, September 2010.

[24] M. Elad, "On the origin of the bilateral filter and ways to improve it," in *IEEE Trans. on Image Processing*, vol. 11, no.10, October 2002, pp. 1141–1151.

[25] W. Sun, G. Cheung, P. Chou, D. Florencio, C. Zhang, and O. Au, “Rate-distortion optimized 3D reconstruction from noise-corrupted multiview depth videos,” in *IEEE International Conference on Multimedia and Expo*, San Jose, CA, July 2013.
[26] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proceedings of the IEEE International Conference on Computer Vision*, Bombay, India, 1998.
[27] H. Takeda, S. Farsiu, and P. Milanfar, “Kernel regression for image processing and reconstruction,” in *IEEE Transactions on Image Processing*, vol. 16, no.3, February 2007, pp. 349–366.
[28] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*, ser. Monographs on Statistics and Applied Probability. London: Chapman & Hall, 2005, vol. 104.
[29] B. Furht, J. Greenberg, and R. Westwater, *Motion estimation algorithms for video compression*, ser. Kluwer Int’l Series in Engineering and Computer Science. Kluwer Acad. Publishers, 1997.
[30] R. Li, B. Zeng, and M.-L. Liou, “A new three-step search algorithm for block motion estimation,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 4, no. 4, pp. 438–442, 1994.
[31] S. Zhu and K.-K. Ma, “A new diamond search algorithm for fast block-matching motion estimation,” *Image Processing, IEEE Transactions on*, vol. 9, no. 2, pp. 287–290, 2000.
[32] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, pp. 671–680, 1983.
[33] N. Jayant, J. Johnston, and R. Safranek, “Signal compression based on models of human perception,” in *Proceedings of the IEEE*, vol. 81, no.10, October 1993, pp. 1385–1422.

Wenxiu Sun received the B.E. degree in Communication Engineering from department of Electronic Engineering, Nanjing University, Nanjing, China in 2009, and the Ph.D. degree in Electronic and Computer Engineering from the Hong Kong University of Science and Technology, Hong Kong in March 2014. In 2012, she was an intern student at National Institute of Informatics (NII), Tokyo, Japan. She currently joined the Image&Visual Computing Lab (IVCL) at Lenovo.

Her research interests include 2D/3D image/video processing techniques, modeling and learning techniques, and their applications in multimedia.



Gene Cheung (M’00—SM’07) received the B.S. degree in electrical engineering from Cornell University in 1995, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California, Berkeley, in 1998 and 2000, respectively.

He was a senior researcher in Hewlett-Packard Laboratories Japan, Tokyo, from 2000 till 2009. He is now an associate professor in National Institute of Informatics in Tokyo, Japan.

His research interests include image & video representation, immersive visual communication and graph signal processing. He has published over 130 international conference and journal publications. He has served as associate editor for IEEE Transactions on Multimedia from 2007 to 2011 and currently serves as associate editor for DSP Applications Column in IEEE Signal Processing Magazine and APSIPA journal on signal & information processing, and as area editor for EURASIP Signal Processing: Image Communication. He currently serves as member of the Multimedia Signal Processing Technical Committee (MMSP-TC) in IEEE Signal Processing Society (2012-2014). He has also served as area chair in IEEE International Conference on Image Processing (ICIP) 2010, 2012-2013, track co-chair in IEEE International Conference on Multimedia and Expo (ICME) 2011, and area chair for ICME 2013. He was invited as plenary speaker for IEEE International Workshop on Multimedia Signal Processing (MMSP) 2013 on the topic “3D visual communication: media representation, transport and rendering”. He is a co-author of best student paper award in IEEE Workshop on Streaming and Media Communications 2011 (in conjunction with ICME 2011), best paper finalists in ICME 2011 and ICIP 2011, best paper runner-up award in ICME 2012, and best student paper award in ICIP 2013.

Philip A. Chou (M'87—SM'00—F'04) received the BSE degree from Princeton University, Princeton, NJ, in 1980, and the MS degree from the University of California, Berkeley, in 1983, both in electrical engineering and computer science, and the PhD degree in electrical engineering from Stanford University in 1988. From 1988 to 1990, he was a Member of Technical Staff at AT&T Bell Laboratories in Murray Hill, NJ. From 1990 to 1996, he was a Member of Research Staff at the Xerox Palo Alto Research Center in Palo Alto, CA. In 1997 he was manager of the compression group at VXtreme, an Internet video startup in Mountain View, CA, before it was acquired by Microsoft in 1997. From 1998 to the present, he has been a Principal Researcher with Microsoft Research in Redmond, Washington, managing the Communication and Collaboration Systems research group from 2004 to 2011. Dr. Chou has served as Consulting Associate Professor at Stanford University 1994-1995, Affiliate Associate Professor at the University of Washington 1998-2009, and Adjunct Professor at the Chinese University of Hong Kong since 2006.

Dr. Chou has longstanding research interests in data compression, signal processing, information theory, communications, and pattern recognition, with applications to video, images, audio, speech, and documents. He served as Associate Editor for the IEEE Transactions on Information Theory and Guest Editor for special issues in the IEEE Transactions on Image Processing, the IEEE Transactions on Multimedia (TMM), and IEEE Signal Processing Magazine. He was a member of the IEEE Signal Processing Society (SPS) Image and Multidimensional Signal Processing technical committee (IMDSP TC) 1998-2004, member and Chair of the SPS Multimedia Signal Processing TC (MMSP TC) 2007-2012, member of the ComSoc Multimedia TC, member of the IEEE SPS Fellow evaluation committee, member of the TMM and ICME Steering Committees, and member of the SPS Board of Governors. He was the founding technical chair for the inaugural NetCod 2005 workshop, special session and panel chair for ICASSP 2007, publicity chair for the Packet Video Workshop 2009, technical co-chair for MMSP 2009, and awards chair for ICIP 2015. He is a Fellow of the IEEE, a member of Phi Beta Kappa, Tau Beta Pi, Sigma Xi, and the IEEE Computer, Information Theory, Signal Processing, and Communications societies, and was an active member of the MPEG committee. He is the recipient, with Tom Lookabaugh, of the 1993 Signal Processing Society Paper Award; with Anshul Seghal, of the 2002 ICME Best Paper Award; with Zhourong Miao, of the 2007 IEEE Transactions on Multimedia Best Paper Award; and with Miroslav Ponec, Sudipta Sengupta, Minghua Chen, and Jin Li, of the 2009 ICME Best Paper Award. He is co-editor, with Mihaela van der Schaar, of the 2007 book from Elsevier, Multimedia over IP and Wireless Networks.

Dinei Florencio (M'97—SM'05) is a researcher with Microsoft Research since 1999, currently with the Multimedia, Interaction and Communication group. He received the B.S. and M.S. from University of Brasilia (Brazil), and the Ph.D. from Georgia Tech, all in Electrical Engineering. Before joining Microsoft, he was a member of the research staff at the David Sarnoff Research Center from 1996 to 1999. He was also a co-op student with AT&T Human Interface Lab (now part of NCR) from 1994 to 1996, and a Summer intern at Interval Research in 1994. He has published over 60 papers, and authored 40 granted patents.

Dr. Florencio is a senior member of the IEEE, and acted as general co-chair of MMSP 2009, Hot3D 2010, and WIFS 2011, and as technical chair of WIFS 2010, ICME 2011, and MMSP 2013. He is the chair of the IEEE SPS Technical Committee on Multimedia Signal Processing (2014-2015). He is also an elected member of the IEEE SPS Technical Committee on Information Forensics and Security, and an associate editor of the IEEE Transactions on Information Forensics and Security.

Cha Zhang (M'01—SM'09) is a senior researcher in the Multimedia, Interaction and Communication Group at Microsoft Research. He received the B.S. and M.S. degrees from Tsinghua University, Beijing, China in 1998 and 2000, respectively, both in Electronic Engineering, and the Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University, in 2004. His current research focuses on applying various audio/image/video processing and machine learning techniques to multimedia applications, in particular, multimedia teleconferencing. Dr. Zhang has published more than 80 technical papers and holds 20+ U.S. patents. He won the best paper award at ICME 2007, the top 10% award at MMSP 2009, and the best student paper award at ICME 2010.

Dr. Zhang is a Senior Member of IEEE. He was the Program Co-Chair for the first Immersive Telecommunication Conference (IMMERSCOM) in 2007, and the Program Co-Chair for VCIP 2012. He currently serves as an Associate Editor for IEEE Trans. on Circuits and Systems for Video Technology, and IEEE Trans. on Multimedia.

Oscar C. Au (M'86—SM'01—F'12) Oscar C. Au received his B.A.Sc. from Univ. of Toronto in 1986, his M.A. and Ph.D. from Princeton Univ. in 1988 and 1991 respectively. After being a postdoc in Princeton for 1 year, he joined Hong Kong Univ. of Science and Technology (HKUST) as an Assistant Professor in 1992. He is/was a Professor of Dept. of Electronic and Computer Engineering, Director of Multimedia Technology Research Center, and Director of Computer Engineering in HKUST. He is a core member of the State Key Lab on Advanced Displays and Optoelectronic Technology.

His main research contributions are on video/image coding and processing, watermarking/light weight encryption, speech/audio processing. Research topics include fast motion estimation for H.261/3/4/5, MPEG-1/2/4, and AVS, optimal and fast sub-optimal rate control, mode decision, transcoding, denoising, deinterlacing, post-processing, multi-view coding, view interpolation, depth estimation, 3DTV, scalable video coding, distributed video coding, subpixel rendering, JPEG/JPEG2000, HDR imaging, compressive sensing, halftone image data hiding, GPU-processing, software-hardware co-design, etc. He has published 65+ technical journal papers, 360+ conference papers, 3 book chapters, and 70+ contributions to international standards.

His fast motion estimation algorithms were accepted into the ISO/IEC 14496-7 MPEG-4 international video coding standard and the China AVS-M standard. His light-weight encryption and error resilience algorithms are accepted into the China AVS standard. He was Chair of Screen Content Coding AdHoc Group in JCTVC for HEVC. He has 25+ granted US patents and is applying for 70+ more on his signal processing techniques. He has performed forensic investigation and stood as an expert witness in Hong Kong courts many times.

Dr. Au is a Fellow of IEEE and HKIE. He is/was Associate/Senior Editors of several IEEE journals (TCSVT, TIP, TCAS1, SPL, JSTSP, SigView) and non-IEEE journals (JVCIR, JSPS, TSIP, JMM, JFI, and SWJ). He is guest editor of some special issues in JSTSP and TCSVT. He is/was BoG member and Vice President Technical Activity of APSIPA. He is/was Chair of 3 technical committees: IEEE CAS MSA TC, IEEE SPS MMSP TC, and APSIPA IVM TC. He is a member of 5 other TCs: IEEE CAS VSPC TC, DSP TC, IEEE SPS IVMSP TC, IFS TC, and IEEE ComSoc MMC TC. He served on 2 steering committees: IEEE TMM, and IEEE ICME. He also served on organizing committee of many conferences including ISCAS 1997, ICASSP 2003, ISO/IEC 71st PCM 2007, ICME 2010, PV 2010, MMSP 2015, APSIPA ASC 2015, and ICME 2017. He won 5 best paper awards: SiPS 2007, PCM 2007, MMSP 2012, ICIP 2013, and MMSP 2013. He was IEEE Distinguished Lecturer (DL) in 2009 and 2010, APSIPA DL in 2013 and 2014, and has been keynote speaker multiple times. MPEG in Jan 2005, ICIP 2010, etc. He was/will be General Chair of several conferences: