JOINT DENOISING / COMPRESSION OF IMAGE CONTOURS VIA GEOMETRIC PRIOR AND VARIABLE-LENGTH CONTEXT TREE

Amin Zheng [*], *Gene Cheung* [#], *Dinei Florencio* [$]

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

ABSTRACT

The advent of depth sensing technologies has eased the detection of object contours in images. For efficient image compression, coded contours can enable edge-adaptive coding techniques such as graph Fourier transform (GFT) and arbitrarily shaped sub-block motion prediction. However, acquisition noise in captured depth images means that detected contours also suffer from errors. In this paper, we propose to jointly denoise and compress detected contours in an image. Specifically, we first propose a burst error model that models typical errors encountered in an observed string $\mathbf{y}$ of directional edges. We then formulate a rate-constrained maximum a posteriori (MAP) problem that trades off the posterior probability $P(\hat{\mathbf{x}}|\mathbf{y})$ of an estimated string $\hat{\mathbf{x}}$ given $\mathbf{y}$ with its code rate $R(\hat{\mathbf{x}})$. Given our burst error model, we show that the negative log of the likelihood $P(\mathbf{y}|\mathbf{x})$ can be written as a simple sum of burst error events, error symbols and burst lengths, while the geometric prior $P(\mathbf{x})$ states intuitively that contours are more likely straight than curvy. We design a dynamic programming (DP) algorithm that solves the posed problem optimally. Experimental results show that our joint denoising / compression scheme outperformed a competing separate scheme in rate-distortion performance noticeably.

Index Terms— Contour coding, joint denoising / compression, image compression

1. INTRODUCTION

Advances in depth sensors like Microsoft Kinect 2.0 means that depth images—per pixel distances between physical objects in a 3D scene and the camera—can now be readily captured. Depth imaging has in turn eased the detection of object contours in a captured image, which was traditionally a challenging computer vision problem [1]. If detected object contours are compressed efficiently as side information, they can enable a plethora of advanced edge-adaptive image coding techniques such as graph Fourier transform (GFT) coding [2, 3] and motion prediction [4] of arbitrarily shaped blocks. Further, coded object contours can also be transmitted to a central cloud for object detection or activity recognition [5], at a much lower coding cost than the original captured depth video.

Unfortunately, captured depth images are often corrupted by acquisition noise, and hence the detected contours also contain errors. We observe that errors in detected contours are unique; *e.g.*, an incorrect depth pixel along a horizontal object boundary will cause the detected contour to go around it, resulting in a *burst error* compared to the original contour. In this paper, we first propose a burst error model that captures unique characteristics of typical errors encountered in detected contours.

This work is supported by the Microsoft Research CORE program.

Moreover, we propose a novel approach to *jointly denoise and compress detected contours* in an image. As done in [6, 7] for multiview depth images, we argue that, given only a noisy observed signal, the signal compression problem is inherently probabilistic. Thus a separate denoising / compression approach—treating a first denoised signal as deterministic input to the second compression stage—cannot perform the best possible rate-distortion (RD) tradeoffs; *e.g.*, a less reliably denoised signal segment should be distorted more (for bit saving) than a more reliably denoised segment.

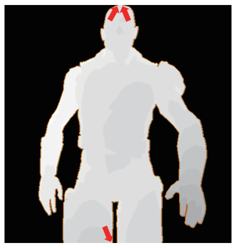
Towards the goal of joint denoising / compression of detected contours, we formulate a rate-constrained *maximum a posteriori* (MAP) problem that trades off the posterior probability $P(\hat{\mathbf{x}}|\mathbf{y})$ of an estimated contour $\hat{\mathbf{x}}$ given observed contour $\mathbf{y}$ with its code rate $R(\hat{\mathbf{x}})$. Given our burst error model, we show that the negative log of the likelihood $P(\mathbf{y}|\mathbf{x})$ can be written as a simple sum of burst error events, error symbols and burst lengths. Further, we construct a geometric prior $P(\mathbf{x})$ stating intuitively that contours are more likely straight than curvy. We design a dynamic programming (DP) [8] algorithm that solves the posed problem optimally in polynomial time. Experimental results show that our joint denoising / compression scheme outperformed a competing separate scheme in RD performance noticeably. We note that, to the best of our knowledge, *we are the first in the literature to formally address the problem of joint denoising / compression of detected image contours*.

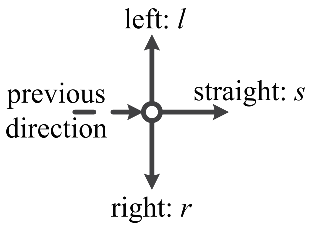
The outline of the paper is as follows. We first overview related works in Section 2. We pose our rate-constrained MAP problem in Section 3, and describe our proposed optimal algorithm in Section 4. Experimental results are presented in Section 5.

2. RELATED WORK

The bulk of existing lossless contour coding research first convert each detected contour into a differential chain code (DCC) [9]—a finite sequence of symbols $\mathbf{x}$ each chosen from a small discrete alphabet $\mathcal{A}$ (more details in Section 3). The conditional probability distribution of a symbol $x_i \in \mathcal{A}$ given D previous symbols $\mathbf{x}_{i-D}^{i-1}$ (*context*) is estimated based on training data. If training data is limited, it has been shown that *variable-length context tree* (VCT) [10], where the context used to define $P(x_i|\mathbf{x}_{i-D}^{i-1})$ may depend only on a sub-string $\mathbf{x}_{i-d}^{i-1}$, $d < D$, avoids over-fitting and has good performance. One of the best VCTs is *prediction by partial matching* (PPM) [11], which we use for actual coding and to compute the rate term $R(\hat{\mathbf{x}})$ in our rate-constrained MAP estimation problem.

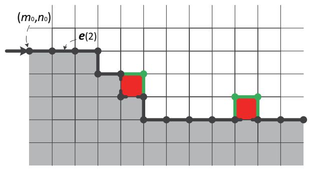
Lossy contour coding approaches can be classified into two categories: DCC-based [12, 13] and vertex-based [14, 15] approaches. In [12, 13], the DCC strings are approximated by using some line smoothing techniques. Vertex-based approaches select representative key points along the contour for coding at the encoder and in-


(a)

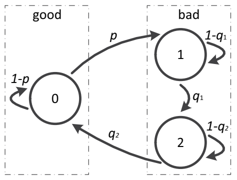


(b)

Fig. 1. (a) Depth image with three detected contours. Initial points of the contours are indicted by red arrows. (b) Directional code.



(a)



(b)

Fig. 2. (a) An example of an observed length-17 contour: $s-s-s-r-l-s-r-s-l-s-s-s-l-r-r-l-s$, and $\mathbf{e}(2) = \{(m_0+2, n_0), E\}$. The two red squares are the erred pixels. Observed $\mathbf{y}$ is composed of black solid edges (good states) and green solid edges (bad states). The ground truth $\mathbf{x}$ is composed of black solid edges and black dotted edges. (b) A three-state Markov model.

terpolation at the decoder. Because vertex-based approaches are not suitable for lossless contour coding and we consider joint denoising / compression of contours for a wide range of bitrates, we choose a DCC-based approach and compute the optimal rate-constrained MAP contour estimate for coding using PPM.

Contour noise removal is considered in [4, 16]. The denoised contour, however, may require a large encoding overhead. In contrast, we perform a MAP estimate to denoise an observed contour subject to a rate constraint. We will show in Section 5 that we outperform a separate denoising / compression approach.

3. PROBLEM FORMULATION

We formulate our rate-constrained MAP problem to find the best contour estimate for lossless coding. We assume that one or more object contours in a noise-corrupted image have first been detected, for example, using a method like gradient-based edge detection [4]. Each contour is defined by an initial point and a following sequence of connected "between-pixel" edges on a 2D grid that divide pixels in a local neighborhood into two sides. As an example, three contours in one frame of `Dude` are drawn in Fig. 1(a).

We first convert each contour into a *differential chain code* (DCC) [9]—a string of symbols each chosen from a size-three alphabet, $\mathcal{A} = \{l, s, r\}$, specifying the three relative directions `left`, `straight`, `right` on the 2D grid, as shown in Fig. 1(b). Denote by $L_{\mathbf{z}}$ the length of DCC string $\mathbf{z}$, where $z_i \in \mathcal{A}$, $1 \leq i \leq L_{\mathbf{z}}$. Denote by $\mathbf{e}_{\mathbf{z}}(i) = \{(m_i, n_i), d_i\}$ the i-th edge of $\mathbf{z}$ on the 2D grid, where (m_i, n_i) is the 2D coordinate of the ending point of the edge, and $d_i \in \mathcal{D} = \{N, E, S, W\}$ is the absolute direction, *i.e.* `North`, `East`, `South` and `West`. Because each contour is processed independently, we can assume that each DCC string starts from the origin, *i.e.*, $(m_0, n_0) = (0, 0)$, as shown in Fig. 2(a).

Denote by $\mathbf{z}_i^j = [z_j, z_{j-1}, \ldots, z_i]$, $i < j$ and $i, j \in \mathbb{Z}^+$, a *sub-string* of length $j - i + 1$ from the i-th symbol z_i to the j-th symbol z_j in reverse order. Denote by $\mathbf{y} \in \mathcal{S}$ and $\mathbf{x} \in \mathcal{S}$ the observed and ground truth DCC strings respectively, where $\mathcal{S}$ is the space of all DCC strings of finite length. As done in [6, 7] for multiview depth images, we follow a rate-constrained MAP formulation and define our objective as finding string $\hat{\mathbf{x}} \in \mathcal{S}$ that maximizes the posterior probability $P(\hat{\mathbf{x}}|\mathbf{y})$ (minimizes $-\log P(\hat{\mathbf{x}}|\mathbf{y})$), subject to a rate constraint on chosen $\hat{\mathbf{x}}$:

$$\min_{\hat{\mathbf{x}} \in \mathcal{S}} \; -\log P(\hat{\mathbf{x}}|\mathbf{y}) \quad \text{s.t. } R(\hat{\mathbf{x}}) \leq R_{\max} \tag{1}$$

where $R(\hat{\mathbf{x}})$ is the bit count required to encode string $\hat{\mathbf{x}}$, and $R_{\max}$ is the bit budget.

Instead of (1), one can solve the corresponding Lagrangian relaxed version instead [17]:

$$\min_{\hat{\mathbf{x}} \in \mathcal{S}} \; -\log P(\hat{\mathbf{x}}|\mathbf{y}) + \lambda R(\hat{\mathbf{x}}) \tag{2}$$

where Lagrangian multiplier λ is chosen so that the optimal solution $\hat{\mathbf{x}}$ to (2) has rate $R(\hat{\mathbf{x}}) \leq R_{\max}$. [17, 18] discussed how to select an appropriate λ. We focus on how (2) is solved for a given λ.

3.1. Likelihood & Prior Terms

Instead of maximizing the posterior $P(\mathbf{x}|\mathbf{y})$, we can maximize the product of likelihood $P(\mathbf{y}|\mathbf{x})$ and prior $P(\mathbf{x})$ using Bayes' Rule. We first describe an error model for DCC strings, then define likelihood $P(\mathbf{y}|\mathbf{x})$ and prior $P(\mathbf{x})$ in turn.

3.1.1. Error Model for DCC String

Assuming that an image is corrupted by a small amount of independent and identically distributed (iid) noise, a detected contour will occasionally be shifted from the true contour by one pixel or two. However, the computed DCC string from the detected contour will experience a sequence of wrong symbols—a *burst error*. This is illustrated in Fig. 2(a), where the first single erred pixel resulted in two erred symbols in the DCC string. The second single error pixel also resulted in a burst error in the observed string, which is *longer* than original string. Based on these observations, we propose our DCC string error model as follows.

We define a three-state Markov model as illustrated in Fig. 2(b) to model the probability of observing DCC string $\mathbf{y}$ given original string $\mathbf{x}$. State `0` is the good state, and *burst error state* `1` and *burst length state* `2` are the bad states. p, q_1 and q_2 are the transition probabilities from state `0` to `1`, `1` to `2`, and `2` to `0`, respectively. Note that state `1` (`0`) cannot transition directly to `0` (`2`).

Starting at good state `0`, each journey to state `1` then to `2` then back to `0` is called a *burst error event*. From state `0`, each self-loop back to `0` with probability $1-p$ means that the next observed symbol y_i is the same as x_i in original $\mathbf{x}$. A transition to burst error state `1` with probability p, and each subsequent self-loop with probability $1 - q_1$, mean observed y_i is now different from x_i. A transition to burst length state `2` then models the *length increase* in observed $\mathbf{y}$ over original $\mathbf{x}$ due to this burst error event: the number of self-loops taken back to state `2` is the increase in number of symbols. A return to good state `0` signals the end of this burst error event.

3.1.2. Likelihood Term

Given the three-state Markov model, we can compute likelihood $P(\mathbf{y}|\mathbf{x})$ as follows. For simplicity, we assume that $\mathbf{y}$ starts and ends at good state 0. Denote by K the total number of burst error events in $\mathbf{y}$ given $\mathbf{x}$. Further, denote by $l_1(k)$ and $l_2(k)$ the number of visits to state 1 and 2 respectively during the k-th burst error event. Similarly, denote by $l_0(k)$ the number of visits to state 0 *after* the k-th burst error event. We can then write the likelihood $P(\mathbf{y}|\mathbf{x})$ as:

$$(1-p)^{l_0(0)} \prod_{k=1}^{K} p(1-q_1)^{l_1(k)-1} q_1 (1-q_2)^{l_2(k)-1} q_2 (1-p)^{l_0(k)-1} \tag{3}$$

For convenience, we define the total number of visits to state 0, 1 and 2 as $\Gamma = \sum_{k=0}^{K} l_0(k)$, $\Lambda = \sum_{k=1}^{K} l_1(k)$ and $\Delta = \sum_{k=1}^{K} l_2(k)$ respectively. We can then write the negative log of the likelihood as:

$$\begin{aligned} &-\log P(\mathbf{y}|\mathbf{x}) = \\ &-K(\log p + \log q_1 + \log q_2) - (\Gamma - K)\log(1-p) \\ &-(\Lambda - K)\log(1-q_1) - (\Delta - K)\log(1-q_2) \end{aligned} \tag{4}$$

Assuming burst errors are rare events, p is small and $\log(1-p) \approx 0$. Hence:

$$\begin{aligned} &-\log P(\mathbf{y}|\mathbf{x}) \\ &\approx -K(\log p + \log q_1 + \log q_2) \\ &-(\Lambda - K)\log(1-q_1) - (\Delta - K)\log(1-q_2) \\ &= -K\underbrace{(\log p + \log \frac{q_1}{1-q_1} + \log \frac{q_2}{1-q_2})}_{-c_0} \\ &-\Lambda \underbrace{\log(1-q_1)}_{-c_1} - \Delta \underbrace{\log(1-q_2)}_{-c_2} \end{aligned} \tag{5}$$

Thus $-\log P(\mathbf{y}|\mathbf{x})$ simplifies to:

$$-\log P(\mathbf{y}|\mathbf{x}) \approx (c_0 + c_2)K + c_1\Lambda + c_2\Delta' \tag{6}$$

where $\Delta' = \Delta - K$ is the length increase of observed $\mathbf{y}$ due to the K burst error events[1]. (6) states that the negative log of the likelihood is a linear sum of three terms: i) the number of burst error events K; ii) the number of error corrupted symbols Λ; and the length increase Δ in observed string $\mathbf{y}$. This agrees with our intuition that more error events, more errors and more deviation in DCC length will result in a larger objective value in (2).

3.1.3. Prior Term

Similar to [4], we propose a *geometric prior* based on the assumption that contours in natural images tend to be more straight than curvy. Specifically, we write prior $P(\mathbf{x})$ as:

$$P(\mathbf{x}) = \exp\left\{-\beta \sum_{i=D_s+1}^{L_\mathbf{x}} s(\mathbf{x}_{i-D_s}^i)\right\} \tag{7}$$

where β is a parameter, and $s(\mathbf{x}_{i-D_s}^i)$ measures the *straightness* of DCC sub-string $\mathbf{x}_{i-D_s}^i$. Let $\mathbf{w}$ be a DCC string of length D_s+1, *i.e.*, $L_\mathbf{w} = D_s + 1$. Then $s(\mathbf{w})$ is defined as the *maximum Euclidean distance* between any coordinates of edge $\mathbf{e}_\mathbf{w}(k), 1 \le k \le L_\mathbf{w}$ and the line connecting the first point (m_0, n_0) and the last point $(m_{L_\mathbf{w}}, n_{L_\mathbf{w}})$ of $\mathbf{w}$ on the 2D grid. We now write $s(\mathbf{w})$ as:

$$\max_{1\le k\le L_\mathbf{w}} \left\{\frac{|(m_k - m_0)(n_{L_\mathbf{w}} - n_0) - (n_k - n_0)(m_{L_\mathbf{w}} - m_0)|}{\sqrt{(m_{L_\mathbf{w}} - m_0)^2 + (n_{L_\mathbf{w}} - n_0)^2}}\right\} \tag{8}$$

Some examples of $s(\mathbf{w})$ are shown in Fig. 3.

[1]Length increase of observed $\mathbf{y}$ due to k-th burst error event is $l_2(k) - 1$.

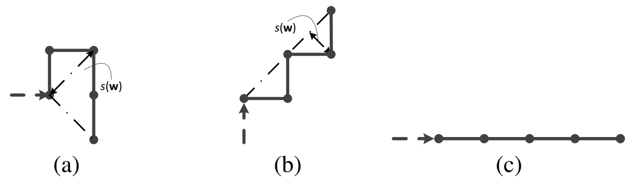


Fig. 3. Three examples of the straightness of $s(\mathbf{w})$ with $L_\mathbf{w} = 4$. (a) $\mathbf{w} = rrls$ and $s(\mathbf{w}) = \sqrt{2}$. (b) $\mathbf{w} = lrlr$ and $s(\mathbf{w}) = \sqrt{2}/2$. (c) $\mathbf{w} = ssss$ and $s(\mathbf{w}) = 0$.

3.2. Rate Term

As discussed, to encode a chosen DCC string $\hat{\mathbf{x}}$ we use PPM [11], which assigns conditional probability $P(\hat{x}_i|\hat{\mathbf{x}}_{i-D_r}^{i-1})$ of symbol $\hat{x}_i \in \mathcal{A}$ given previous D_r symbols $\hat{\mathbf{x}}_{i-D_r}^{i-1}$. The computed probabilities are input to an arithmetic coder for entropy coding. The number of bits for coding $\hat{\mathbf{x}}$ are thus approximated as the negative log of conditional probabilities:

$$R(\hat{\mathbf{x}}) = -\sum_{i=D_r+1}^{N} \log_2 P(\hat{x}_i|\hat{\mathbf{x}}_{i-D_r}^{i-1}) \tag{9}$$

Having defined the likelihood, prior and rate terms, our Lagrangian objective (2) can now be rewritten as:

$$\begin{aligned} J(\hat{\mathbf{x}}) &= (c_0 + c_2)K + c_1\Lambda + c_2\Delta' \\ &+ \beta \sum_{i=D_s+1}^{L_{\hat{\mathbf{x}}}} s(\hat{\mathbf{x}}_{i-D_s}^i) - \lambda \sum_{i=D_r+1}^{L_{\hat{\mathbf{x}}}} \log_2 P(\hat{x}_i|\hat{\mathbf{x}}_{i-D_r}^{i-1}) \end{aligned} \tag{10}$$

4. OPTIMIZATION ALGORITHM

We present our DP algorithm to solve (10) optimally. To simplify the notation, we define $D = \max\{D_s, D_r\}$ and

$$f(\hat{\mathbf{x}}_{i-D}^i) = \beta \sum_{i=D_s+1}^{L_{\hat{\mathbf{x}}}} s(\hat{\mathbf{x}}_{i-D_s}^i) - \lambda \sum_{i=D_r+1}^{L_{\hat{\mathbf{x}}}} \log_2 P(\hat{x}_i|\hat{\mathbf{x}}_{i-D_r}^{i-1}). \tag{11}$$

Further, we let the first D estimated symbols $\hat{\mathbf{x}}_1^D$ be the observed $\mathbf{y}_1^D$, and assume the last edge is correct, *i.e.*, $\mathbf{e}_{\hat{\mathbf{x}}}(L_{\hat{\mathbf{x}}}) = \mathbf{e}_\mathbf{y}(L_\mathbf{y})$.

In a nutshell, as we examine each symbol in observed $\mathbf{y}$, we identify an "optimal" state traversal through our 3-state Markov model—one that minimizes objective (10)—via two recursive functions. The optimal state traversal translates directly to an estimated DCC string $\hat{\mathbf{x}}$, which is the output of our algorithm.

Denote by $G_i(\hat{\mathbf{x}}_{i-D}^{i-1}, \mathbf{e}, j-1)$ the minimum cost for estimated $\hat{\mathbf{x}}$ from the i-th symbol onwards, given that we are in the good state with a set of D previous symbols (context) $\mathbf{x}_{i-D}^i$, last edge being $\mathbf{e}$ which is the same as an edge of index $j-1$ in observed $\mathbf{y}$. If we choose one additional symbol $\hat{x}_i = y_j$, then we remain in the good state, incurring a local cost $f(\hat{\mathbf{x}}_{i-D}^i)$, plus a recursive cost $G_{i+1}(\,)$ for the remaining symbols in string $\hat{\mathbf{x}}$ due to new context $\hat{\mathbf{x}}_{i-D+1}^i$.

If instead we choose one additional symbol $\hat{x}_i \ne y_j$, then we start a new burst error event, incurring a local cost $(c_0 + c_2)$ for the new event, in addition to $f(\hat{\mathbf{x}}_{i-D}^i)$. Entering bad states, we use $B_{i+1}(\,)$ for recursive cost instead.

$B_i(\hat{\mathbf{x}}_{i-D}^{i-1}, \mathbf{e}, j-1)$ is similarly computed as $G_i(\hat{\mathbf{x}}_{i-D}^{i-1}, \mathbf{e}, j-1)$, except that if chosen symbol $\hat{x}_i$ has no corresponding edge in $\mathbf{y}$, then we add c_1 to account for an additional error symbol. If chosen

$$G_i(\hat{\mathbf{x}}_{i-D}^{i-1}, \mathbf{e}, j-1) = \min_{\hat{x}_i \in \mathcal{A}} \begin{cases} f(\hat{\mathbf{x}}_{i-D}^{i}) \;+\; \mathbf{1}(j < L_{\mathbf{y}})\; G_{i+1}(\hat{\mathbf{x}}_{i-D+1}^{i}, v(\mathbf{e}, \hat{x}_i), j), & \text{if } \hat{x}_i = y_j \\ (c_0 + c_2) + f(\hat{\mathbf{x}}_{i-D}^{i}) \;+\; B_{i+1}(\hat{\mathbf{x}}_{i-D+1}^{i}, v(\mathbf{e}, \hat{x}_i), j), & \text{o.w.} \end{cases} \tag{12}$$

$$B_i(\hat{\mathbf{x}}_{i-D}^{i-1}, \mathbf{e}, j-1) = \min_{\hat{x}_i \in \mathcal{A}} \begin{cases} c_2(k-j) \;+\; f(\hat{\mathbf{x}}_{i-D}^{i}) \;+\; G_{i+1}(\hat{\mathbf{x}}_{i-D+1}^{i}, v(\mathbf{e}, \hat{x}_i), k), & \text{if } \exists k, v(\mathbf{e}, \hat{x}_i) = \mathbf{e}_{\mathbf{y}}(k) \\ c_1 \;+\; f(\hat{\mathbf{x}}_{i-D}^{i}) \;+\; B_{i+1}(\hat{\mathbf{x}}_{i-D+1}^{i}, v(\mathbf{e}, \hat{x}_i), j), & \text{o.w.} \end{cases} \tag{13}$$

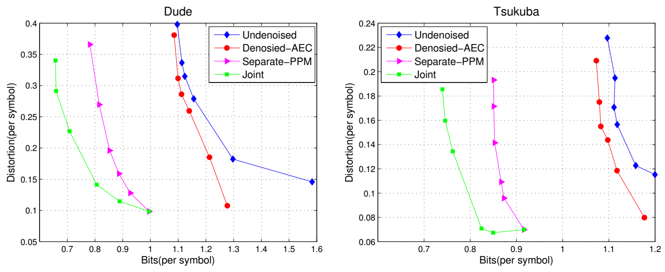


Fig. 4. Rate distortion curve of `Dude` and `Tsukuba`

symbol $\hat{x}_i$ has a corresponding edge in $\mathbf{y}$, then this is the end of the burst error event, and we return to good state (recursive call to $G_{i+1}(\,)$ instead). In this case, we must account for the change in length between $\hat{\mathbf{x}}$ and $\mathbf{y}$ due to this burst error event, weighted by c_2. $G_i(\hat{\mathbf{x}}_{i-D}^{i-1}, \mathbf{e}, j-1)$ and $B_i(\hat{\mathbf{x}}_{i-D}^{i-1}, \mathbf{e}, j-1)$ are defined in (12) and (13), respectively. $\mathbf{1}(\mathbf{c})$ is an indicator function that evaluates to 1 if the specified binary clause $\mathbf{c}$ is true and 0 otherwise. $\mathbf{e} = \{(m,n), d\}$ denotes the $(i-1)$-th edge of $\hat{\mathbf{x}}$, and $v(\mathbf{e}, \hat{x}_i)$ denotes the next edge given that the next symbol is $\hat{x}_i$. $j-1$ is the index of matched edge in $\mathbf{y}$.

In practice, the denoised DCC string $\hat{\mathbf{x}}$ should be no longer than the observed $\mathbf{y}$, so if $i > L_{\mathbf{y}}$ or $k - j < 0$, we stop the recursion and return infinity to signal an invalid solution.

The complexity of the DP algorithm is bounded by the size of the DP tables times the complexity of computing each table entry. Denote by Q the number of possible edge end point locations. Looking at the arguments of the two recursive funcions $G_i(\,)$ and $B_i(\,)$, we see that DP table size is $O(3^D\, 4Q\, L_{\mathbf{y}}^2)$. To compute each entry in $B_i(\,)$, for each $\hat{\mathbf{x}}_i \in \mathcal{A}$, one must check for matching edge in $\mathbf{y}$, hence the complexity is $O(3L_{\mathbf{y}})$. Hence the total complexity of the algorithm is $O(3^D Q L_{\mathbf{y}}^3)$, which is polynomial time for a fixed D.

5. EXPERIMENTATION

We used two computer-generated (noiseless) depth sequences: `Dude` (800×400) and `Tsukuba` (640×480). We used gradient-based edge detection [4] to detect the ground truth contours, and then injected noise to the depth images assuming that the pixels along the contours were corrupted by iid noise: for each edge of the contour, the corruption probability was set at 10%. If corrupted, the pixel along one side of this edge was replaced by the pixel from the other side (side was chosen with equal probability). The noisy contours were then detected from the noisy depth images.

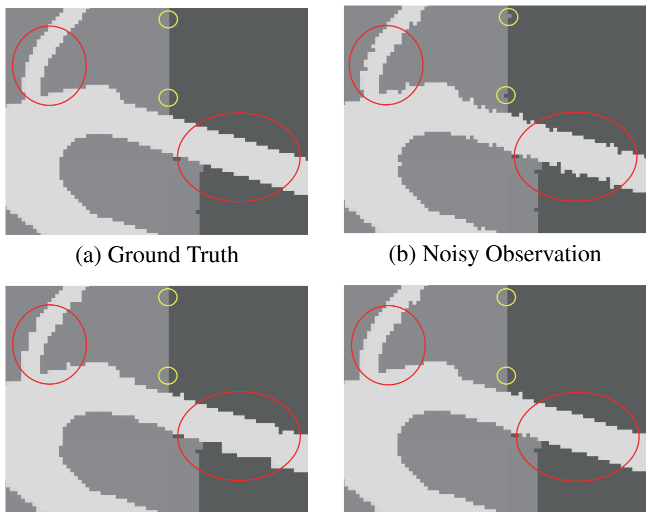
The three transition probabilities in our error model, p, q_1 and q_2, were computed from average noise statistics. Contours in one previous frame were used to train the context tree for PPM-based contour coding. We set $D_s = 4$ and $D_r = 6$ in all the experiments.

We compared performance of four different schemes. The first scheme, `Undenoised`, encoded the noisy contours $\mathbf{y}$ directly using a lossy contour coding method (`Lossy-AEC`) [13]. The second scheme, `Denoised-AEC`, first denoised the contours using an irregularity-detection method [4], then used `Lossy-AEC` for lossy contour coding. The third scheme, `Separate-PPM`, first denoised the contours using our proposal by setting $\lambda = 0$, then used PPM to encode the denoised contours. The fourth scheme, `Joint`, is our proposal that performed joint denosing / compression of contours.

We show the RD performance of the four schemes in Fig. 4. Distortion is defined as the sum of minimum distance from the points on each decoded contour to the corresponding ground truth contour. For `Undenoised` and `Denoised-AEC`, RD-curves were generated by adjusting the strength of contour approximation [13], while the RD-curve for `Joint` was obtained by varying λ. For `Separate-PPM`, we used the same fixed β as in `Joint` to get the best denoising performance, then we increased β to over-smooth the contour to reduce the bit rate and obtained the RD-curve.

We see that `Joint` achieved the best RD performance for both sequences, demonstrating the merit of our joint approach. In particular, we save about 18%, 40% and 45% bits on average against `Separate`, `Denoised-AEC` and `Undenoised` for `Dude`, and about 11%, 31%, 36% for `Tsukuba`.

Fig. 5 illustrates some visual denoising results. We see that the noisy pixels along both the vertical (yellow circles) and diagonal contours (red circles) were mostly removed by our proposed scheme, while `Denoised-AEC` failed to remove the noisy pixels along the diagonal contours.



(a) Ground Truth (b) Noisy Observation

(c) `Denoised-AEC` (d) `Joint`

Fig. 5. Visual denoising results of `Tsukuba`. (c) `Denoised-AEC` at bit rate 1.18 bits/symbol. (d) `Joint` at bit rate 0.92 bits/symbol.

6. REFERENCES

[1] C. Grigorescu, N. Petkov and M. A. Westenberg, “Contour de-

tection based on nonclassical receptive field inhibition," *IEEE Trans. Image Process.*, vol. 12, no. 7, pp. 729–739, 2003.

[2] W. Hu, G. Cheung, A. Ortega, and O. Au, "Multi-resolution graph Fourier transform for compression of piecewise smooth images," in *IEEE Transactions on Image Processing*, January 2015, vol. 24, no.1, pp. 419–433.

[3] W. Hu, G. Cheung, and A. Ortega, "Intra-prediction and generalized graph Fourier transform for image coding," in *IEEE Signal Processing Letters*, November 2015, vol. 22, no.11, pp. 1913–1917.

[4] I. Daribo, D. Florencio, and G. Cheung, "Arbitrarily shaped motion prediction for depth video compression using arithmetic edge coding," in *IEEE Transactions on Image Processing*, November 2014, vol. 23, no. 11, pp. 4696–4708.

[5] D. Weinland, R. Ronfard and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer Vision and Image Understanding*, vol. 115, no. 2, pp. 224–241, 2011.

[6] 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.

[7] W. Sun, G. Cheung, P. Chou, D. Florencio, C. Zhang, and O. Au, "Rate-constrained 3D surface estimation from noise-corrupted multiview depth videos," in *IEEE Transactions on Image Processing*, July 2014, vol. 23, no.7, pp. 3138–3151.

[8] S. E. Dreyfus and A. M. Law, *Art and Theory of Dynamic Programming*, Academic Press, Inc., 1977.

[9] H. Freeman, "Application of the generalized chain coding scheme to map data processing," 1978, pp. 220–226.

[10] R. Begleiter, R. El-Yaniv and G. Yona, "On prediction using variable order markov models," *Journal of Artificial Intelligence Research*, pp. 385–421, 2004.

[11] J. Cleary and I. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Trans. Commun.*, vol. 32, no. 4, pp. 396–402, 1984.

[12] S. Zahir, K. Dhou and B. Prince George, "A new chain coding based method for binary image compression and reconstruction," in *Proc. IEEE Int. Picture Coding Symp.*, 2007, pp. 1321–1324.

[13] Y. Yuan, G. Cheung, P. Frossard, P. L. Callet and V. Zhao, "Contour approximation & depth image coding for virtual view synthesis," in *Proc. IEEE Workshop Multimedia Signal Processing*. IEEE, 2015, pp. 1–6.

[14] A. K. Katsaggelos, L. P. Kondi, F. W. Meier, W. Fabian, J. O. Ostermann and G. M. Schuster, "MPEG-4 and rate-distortion-based shape-coding techniques," *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1126–1154, 1998.

[15] Z. Lai, J. Zhu, Z. Ren, W. Liu and B. Yan, "Arbitrary directional edge encoding schemes for the operational rate-distortion optimal shape coding framework," in *Proc. Conf. Data Compression*. IEEE, 2010, pp. 20–29.

[16] D. Yu and H. Yan, "An efficient algorithm for smoothing, linearization and detection of structural feature points of binary image contours," *Pattern Recognition*, vol. 30, no. 1, pp. 57–69, 1997.

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

[18] A. De Abreu, G. Cheung, P. Frossard, and F. Pereira, "Optimal Lagrange multipliers for dependent rate allocation in video coding," in *arXiv:1509.02995 [cs.MM]*, March 2016.