LOSSY AND NEAR-LOSSLESS COMPRESSION OF DEPTH IMAGES USING SEGMENTATION INTO CONSTRAINED REGIONS

Ionut Schiopu and Ioan Tabus

Department of Signal Processing, Tampere University of Technology,
P.O.Box 553, FIN-33101 Tampere, FINLAND
ionut.schiopu@tut.fi and ioan.tabus@tut.fi

ABSTRACT

This paper presents a lossy coding method for depth images using a segmentation constructed by selecting regions of pixels having the depth values obeying constraints defined in terms of some bounds, which are tuned in order to obtain the target distortion. The contours describing the segmentation are transmitted using an efficient chain coding method and are thus available also at the decoder for the next stage, which is region based predictive coding with a tunable precision level. The rate comprises the cost of losslessly transmission of the contours and the cost of transmitting the residuals with the decided precision, which is the main factor influencing the distortion. We introduce a procedure optimizing the parameters involved in the segmentation and in the prediction for a given image. As a side result, the segmentations residing on the convex hull of the RD curve can be seen as optimal segmentations with various granularity.

Index Terms— lossy compression, near-lossless compression, depth image, segmentation, rate-distortion

1. INTRODUCTION

The subject of depth compression has received increased attention recently, mostly due to the wide range of applications for 3D representations, in computer vision, 3DTv, and computer games. Applying the same compression methods for depth images as for the gray-level or color pictures is not as efficient as designing new methods, dedicated to the types of regularities present in depth images. In [1] we have shown that dedicated lossless compression methods can reduce to half the size of compressed files produced by the standard JPEG-LS image compressor.

The literature on lossy depth image compression is wide, mostly in connection to the compression of multiview images, where the interesting bit-rates are in the very low end, even below 0.1 bpp, see e.g. [2][3] and reference lists therein. Our method addresses higher rate ranges aiming at near-lossless coding, with prior work existing in, e.g., [4] [5].

2. THE PRINCIPLE OF THE METHOD

The basic principle of constructing the segmentation is to split the image into two types of regions: for the first type, a region ”with local variability λ” should contain any pixel which has inside the region at least one neighbor such that the difference between the depth values of the pixel and its neighbor is at most equal to a given threshold λ, and additionally the size of the region is also constrained. In the case of regions of second type, from the regions with local variability $\lambda = 1$ are selected those which have also global variability 1, i.e., they contain pixels with only two distinct depth values. The regions obeying the local constraint variability condition may have quite a diverse distribution of depth values, e.g., planar sections starting with one side close to the camera, with a low depth level and ending on the other side with a very high depth value; as a different example the regions are encompassing also second order surfaces, typical in the case of round, spherical, cylindrical, or conical objects. By varying the threshold λ the sizes of the regions will change. The process of finding the regions is iterative, starting with finding the connected regions at small thresholds and if they are large enough they are declared regions, and then the process continues at larger thresholds. The values of the thresholds and the lower bound of the region size for declaring a region are parameters determining many possible partitions of the image into regions. When the parameters determine a rough granularity, the cost of transmitting the contours of regions is small. Here we use chain codes which are very cheap ways of transmitting losslessly the contours, and hence we do not resort to parametric models for coding the contours in a lossy manner, which is the option followed in most of the previous lossy coding methods.

After the regions are defined, in each region we use lossy predictive coding, where the prediction is performed based on the reconstructed depth of the previously transmitted pixels and the quantization of the prediction residuals is uniform, with a tunable step size $2\eta + 1$, for all regions except those regions which contain two or three distinct depth values. The parameter η defining the quantization steps belongs to the set $\{0, 1, 2, 3\}$, where $\eta = 0$ means no quantization, in which case the compression is lossless for the involved regions. The regions having only two or three distinct depth values are treated differently; in each of them only one reconstruction value is chosen (the one minimizing the sum of square errors inside the region) and then is transmitted to the decoder.

When the targeted bitrate is in the low range (below 0.2 bpp), we introduce an additional preprocessing stage: before segmentation stage the last bit of the depth values is removed, the average of all these removed bits is computed, and if the average is larger than 0.5, a bit of one is appended as a least significant bit of the final depth values obtained after the reconstruction at the decoder. In this way the range of depth values is halved and only a rough reconstruction of the last bit is performed, by the majority bit.

In the following section we present the algorithmic design of the segmentation and the way to combine the lossless contour compression and the lossy prediction residuals encoding. The obtained results are compared with [4].

The depth image contains the depth value $I(x,y) \in \{0, 2^B - 1\}$ for each pixel (x,y). For illustrations we use as input image the same one used in [4], namely the first frame of view 1 from the *Breakdancing* sequence [6], which has the number of bit-planes $B = 8$. We will define the segmentation as the union of all regions, $\Omega_1, \ldots, \Omega_{n_r}$, that make up the depth image.

3. ALGORITHM DESCRIPTION

3.1. Generating a segmentation

The main problem for obtaining the best results is to generate a suitable segmentation of the image so that after lossless contour compression the decoder knows enough distinct regions and with an additional bitstream containing the prediction residuals it can obtain small overall distortion using a low bitrate. Our solution is an iterative segmentation which allows a different variability inside some regions, while for other regions, beside imposing the fixed variability $\lambda = 1$, it is additionally required that the overall number γ of distinct depth-levels is small. At a given step of the algorithm all connected components which contain more than N_λ (or K_γ) pixels are declared regions. The process can be characterized by the maximum allowed variability λ with its associated minimum size N_λ of a region, and the constrained number γ of distinct depth-levels with its associated minimum size K_γ.

For a current pixel (x_t, y_t) the set of the four neighbors in 4-connectivity is denoted $\mathcal{N}_4(x_t, y_t)$. The variability of the current pixel (x_t, y_t) inside a given region Ω_j is defined as the minimum value of the absolute differences of the depth values between the current pixel and those neighbors which are part of the region [1], as follows:

$$V(x_t, y_t) = \min_{(x_i,y_i)\in\mathcal{N}_4(x_t,y_t)\cap\Omega_j} |I(x_t, y_t) - I(x_i, y_i)|.$$

The segmentation algorithm starts by finding the sets of connected pixels having variability $\lambda = 0$ and declaring them candidate regions. In the next stage each candidate region having at least K_γ pixels is declared a region Ω_j of the segmentation, and the remaining pixels, not yet in the already decided regions, are then grouped in connected components that have variability at most $\lambda = 1$ and a number of distinct depth-levels $\gamma = 2$. Using these constraints one can generate regions that have only two consecutive depth-levels that are compressed using a single reconstruction depth level, the one which gives the minimum distortion. In the next step we iterate for successive thresholds on variability $\lambda \in \Lambda = \{1, 3\}$. At each iteration step, each candidate region from the previous step having at least N_λ pixels is declared a region Ω_j of the segmentation, and the remaining pixels not yet in already decided regions are then grouped in connected components and all such components with variability not exceeding λ are declared candidate regions for the next step. At the last step all candidate regions are automatically declared regions of the segmentation. According to the definition of the variability constrained regions, the obtained segmentation is unique. The regions with size smaller than 5 pixels, which make a large proportion of the whole number of regions, are merged with the largest neighbor region because of the high cost of transmitting them separately. This ensures the reduction of the contour length, and even more importantly, the elimination of some points in the contour with more than two contour-edge intersections, points that are required to be transmitted separately as anchor points and which require a large number of bits for encoding.

We consider in the experiments two versions of the segmentation, the first L-CRS using the merging of the very small regions, the second, NL-CRS keeping the small regions as part of the segmentation.

3.2. Quantization and encoding of regions with almost constant depth

The obtained regions which have a small number, γ, of distinct levels are quantized and encoded in a simpler manner than the rest of the regions. This simple quantization and encoding is used because a near-lossless quantization could have set the regions with a depth-level that produces a large distortion.

In each region that has $\gamma = 2$ distinct depth levels, g and $g + 1$, the quantized depth value is set to that value, g^*, which occur the most often among the two consecutive levels, this process being equivalent to the reconstruction minimizing the distortion.

In each region that has $\gamma = 3$ distinct depth levels, first the mean square distortion after quantizing by the optimal level is computed, and if the resulting PSNR is smaller than a threshold T_3 then the region is quantized and encoded using the predictive method presented in the next section, otherwise it

d	c	
a	x	

Table 1. The prediction neighborhood $\mathcal{N}_P$ of the current pixel (x_t, y_t), which is marked by "x". Also shown are the letters used for the depth values of the neighbors.

is processed similarly to the case $\gamma = 2$, where only the optimal quantization level is encoded and used as a reconstruction at the decoder.

3.3. Local nonlinear prediction

We predict the depth $I(x_t, y_t)$ for a current pixel $(x_t, y_t) \in \Omega$ by using the reconstructed values $\tilde{I}(x_i, y_i)$ of the pixels $(x_i, y_i) \in \Omega$ which also belong to a causal neighborhood $\mathcal{N}_P(x_t, y_t)$ of the pixel (x_t, y_t), depicted in Table 1. We denoted by $\tilde{I}(x, y)$ the value available at the decoder, obtained using the quantized prediction residuals. Similarly as in [1], for each region the horizontal scanning and the vertical scanning are tested, both with causal neighbors (a, c, d), and the one giving better performance is selected and announced to the decoder by one bit per region. Although both scanning orders use the same causal neighborhood, different quantized residues are obtained for each scanning order and hence the two compression ratios for a region are different.

In [1] we used an optimal predictor selected among 15 mixture predictors, $\hat{I}_n(\mathcal{N}_P(x_t, y_t)), n = 1, \ldots, 15$. Here the optimal predictor is taken the one with index $n = 1$, which gave the best results in our tests. Hence, the collection of elementary predictions of the nonlinear predictor, denoted $\mathcal{P}(\mathcal{N}(x_t, y_t))$, is $\mathcal{P}(\mathcal{N}(x_t, y_t)) = \{a,\ c,\ c + a - d\}$. If one of the used neighbors are not in the causal neighborhood, the elementary prediction is eliminated from $\mathcal{P}(\mathcal{N}(x_t, y_t))$.

Consequently, in this paper the prediction is calculated as follows: $\hat{I}_n(\mathcal{N}(x_t, y_t)) = median\{\mathcal{P}_n(\mathcal{N}(x_t, y_t))\}$.

3.4. Encoding of quantized prediction residuals

The encoding of the pixels in any region Ω_i having k_i pixels is performed as follows. First determine the prediction value $\hat{I}(x_t, y_t)$ for each pixel (x_t, y_t) in the region, with $t = 1, \ldots, k_i$. We define the residuals $\epsilon(x_t, y_t) = I(x_t, y_t) - \hat{I}(x_t, y_t)$ for all pixels $(x_t, y_t) \in \Omega_i$. In the next step we quantize the prediction residuals $\epsilon(x_t, y_t)$ using uniform quantization. Same as in [7], the quantizer is defined as:

$$Q(\epsilon) = sign(\epsilon) \left\lfloor \frac{|\epsilon| + \eta}{2\eta + 1} \right\rfloor,$$

where the signum function returns 1 for positive argument, -1 for negative and 0 for 0 argument. The reconstructed value, used also by the encoder, is obtained as follows:

$$\tilde{I} = \hat{I} + Q'(\epsilon) \cdot (2\eta + 1),$$

where the reconstruction level is biased from the midpoint of the quantization interval towards zero, to account for the typical monotonic decreasing pdf of the absolute value of the residual:

$$Q'(\epsilon) = Q(\epsilon) - \frac{sign(Q(\epsilon)) \cdot \mu(\eta)}{2\eta + 1}.$$

The tests showed that the best results are obtained for the bias term corresponding to $\mu(\eta) = \eta$.

In the next stage we determine the minimum and maximum quantized residuals, denoted by m_i and M_i for each $\Omega_i \in I$ and encode them along with all auxiliary information. We form a stream of symbols by concatenating the shifted residuals $\varepsilon'(x_t, y_t) = Q(\epsilon(x_t, y_t)) - m_i$ for all regions and encode it by applying adaptive Markov arithmetic coding with order one. Like in [1], for a better compression, when a shifted quantized residual $\varepsilon'(x_t, y_t)$ is larger than an optimally determined value, M_{Res}, we encode the sequence $\{M_{Res}, s, r\}$, where s and r are the quotient and remainder of the division $\frac{\varepsilon'(x_t, y_t)}{M_{Res}}$, respectively.

3.5. Encoding of region contours

The segmentation of an image is defined by contours separating the regions. The contour is transmitted using the 3OT chain-code representation [8].

From the five options presented in [1], option four obtained the best results in our tests. For other images the optimal option could be any other. The fourth option was encoding the chain code by applying the Arithmetic Coding Algorithm using the optimal context tree obtained by the *Context Tree Algorithm* [8]. Depending on the required bitstream, the algorithm generates a specific segmentation which has a different contour. For each segmentation we usually obtain a different tree-depth for the context tree: a height tree-depth ($18 \div 20$) for high bitrate (almost lossless), and a lower tree-depth ($14 \div 16$) for low bitrate, see column 5 of Table 2.

3.6. Summary of the algorithm

The segmentation algorithm implemented, denoted Lossy Constrained Region Segmentation (L-CRS), can be summarized in a few steps:

1. Smooth the contour by eliminating some edges between pixels with similar values as follows: if at least 3 neighbors in $\mathcal{N}_4(x_t, y_t)$ have the same depth value, v, and if $|I(x_t, y_t) - v| \leq 2$, then set $I(x, y) = v$;
2. Determine the connected sets of pixels having variability $\lambda = 0$ (thus having constant depth);
3. Select the sets of pixels with cardinality larger than K_2 and declare them regions; out of the remaining pixels, determine connected sets of pixels having variability at most $\lambda = 1$ and having the maximum number of distinct depth-values $\gamma = 2$;

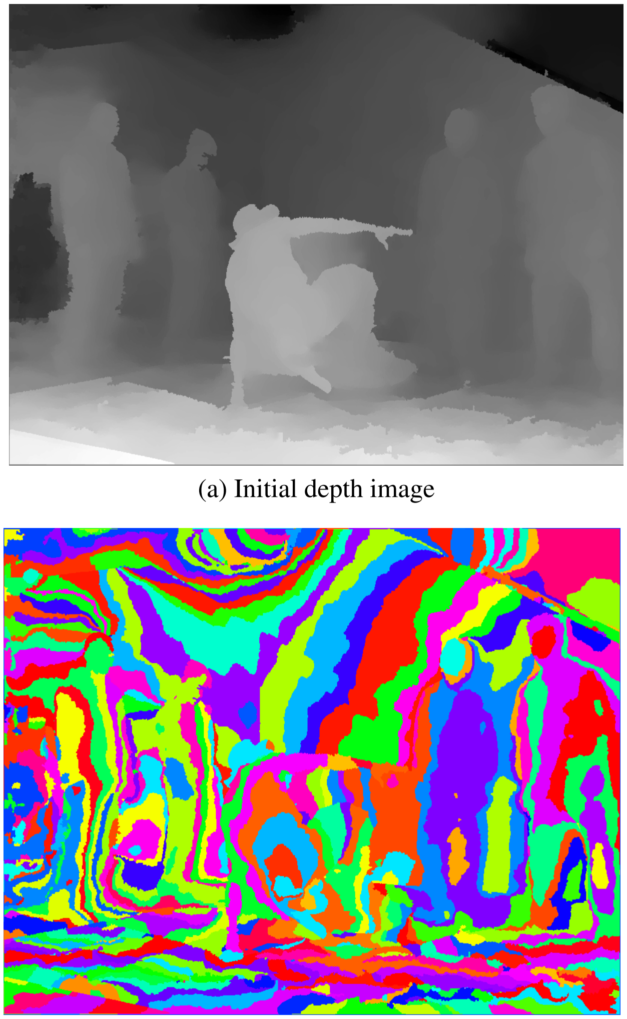
4. Select the sets of pixels with cardinality larger than N_1 and declare them regions; out of the remaining pixels, determine connected sets of pixels having variability at most $\lambda = 1$;
5. Similar to (3), but with cardinality larger than $N_3 = 100$ and using variability $\lambda = 3$;
6. Declare the remaining connected sets of pixels regions;
7. Set pixel regions with size smaller than 5 pixels with the depth level of the biggest neighboring region;
8. Encode the region contours using 3OT chain-codes;
9. Quantize and encode the regions with almost constant depth, using $T_3 = 50$ ($T_3 = 45$ if $low\_bitrate = 1$);
10. For the remaining regions use the near-lossless predictive compression as explained in Sections 3.3 and 3.4.

Besides the fixed parameters, having the value specified in the algorithm, the values of the parameters M_2 and N_1 are the most important and have to be chosen according to the desired bitrate. Both of them can take values from 50 to the size of the biggest region in the image. For example to obtain a compression with a large PSNR, one can set $M_2 = 50$, $N_1 = 50$ and use $eta = 0$, so that only a few regions contains distortion, while for a low bitrate one can set $low\_bitrate = 1$, $M_2 = 1000$, $N_1 = 14000$ and use $eta = 2$.

For a near-lossless compression, meaning that the absolute value of the error is smaller or equal than $2\eta + 1$, we introduced another method, denoted Near-Lossless Constrained Region Segmentation (NL-CRS), using the same algorithm as L-CRS with the difference that we eliminate step (7).

3.7. Experimental results

The segmentation algorithm is illustrated by presenting the segmentation result in Figure 1 (b) for the depth image from Figure 1 (a). We note that the image is oversegmented, in the sense that the meaningful objects are split into many regions, since this split is optimal for our rate-distortion optimization scheme.



(a) Initial depth image

(b) Obtained segmentation

Fig. 1. Example of segmentation for a low bitrate of first frame of view 1 of Breakdancing sequence.

We present the results in a rate-PSNR plot, where the bitrate is calculated as bits per pixel (*bpp*),

$$bpp = 8 \cdot \frac{compreesed\_file\_size}{initial\_file\_size},$$

and the peak signal-to-noise ratio, $PSNR$, is computed as:

$$PSNR = 10 \cdot \log_{10} \frac{255^2}{MSE}.$$

We compared the results for the two methods introduced, L-CRS and NL-CRS, with JPEG2000 and the two other methods from [4] and [5], denoted here Method 1 and Method 2. Figure 2 shows the results for the five methods using one image from the Breakdancing sequence. One can see that our methods obtain better results compared with the best existing results, Method 1 [4]. Because L-CRS is generated from a lossless method, the transition from lossless to lossy is steep. Another factor is that the bitrate has two parts: we compressed losslessly the region contours and lossy the depth-levels inside the regions, that is why for a low bitrate the results will asymptotically reach the point where most of the bitrate will be composed of contour lossless bitrate.

Figure 2 presents also the result of lossless compression using the more complex algorithm from [1]. The result is presented using a vertical asymptote at $bitrate = 0.3933$ *bpp*, which is the point where $PSNR = \infty$.

In Table 2 we take a closer look at some statistics of the segmentation and the proportions of bitrates needed for lossless compression of contours and lossy compression of depth values for 8 functioning points from the L-CRS curve pre-

	Image Segmentation				Q.	LP	Contour comp.		Depth-level comp.		L-CRS	
Nr.	low	final nr.	contour	maximum	η		bitrate	% of	bitrate	% of	Total	PSNR
	bitrate	of reg.	length	tree-depth			(bpp)	total	(bpp)	total	bitrate	(db)
1	0	2575	142949	18	0	461	0.2265	64.48	0.1248	35.52	0.3513	61.9048
2	0	1635	140289	20	1	579	0.2223	82.80	0.0462	17.20	0.2685	57.9007
3	0	1483	134155	20	1	568	0.2120	84.39	0.0392	15.61	0.2512	56.9242
4	0	1269	124382	18	1	474	0.1957	84.86	0.0349	15.14	0.2307	55.3445
5	0	1133	112289	18	2	477	0.1775	88.31	0.0235	11.67	0.2010	53.1608
6	1	598	80145	18	1	193	0.1223	91.37	0.0116	8.63	0.1339	48.2914
7	1	536	70201	18	1	147	0.1078	92.52	0.0087	7.48	0.1166	46.7343
8	1	432	62313	17	3	147	0.0952	93.19	0.0070	6.81	0.1022	45.0466

Table 2. Examples of functioning points on the L-CRS curve from the rate-PSNR plot, with their details regarding the segmentation, the quantization, and the lossy and lossless composition of compressed image. The size of the images is 1024×768 and has 6690 initial constant regions.

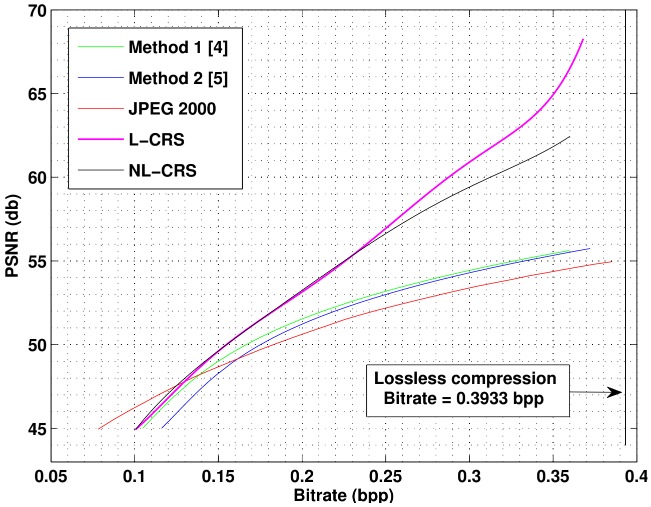


Fig. 2. Lossy depth image compression comparison of our two implemented methods, L-CRS and NL-CRS, with JPEG 2000 and the two methods from [4] and [5], for the first frame of view 1 (camera 0) of the Breakdancing sequence.

sented in Figure 2. The table presents also the number of pixels, denoted lossy pixels (LP), for which we do not impose a near-lossless compression which means that the absolute value between initial depth-level and the reconstructed value is greater than $2\eta + 1$.

In Figure 1 (b) we presented the segmentation obtained for the 6^{th} point in Table 2. One can see that some of the 598 final regions are very small, especially in the bottom of the image. The final segmentation contains these regions because the depth has a great variation in this area and putting them together produces worse results, by increasing significantly the distortion, with just a small decrease of the bitrate.

4. REFERENCES

[1] I. Schiopu and I. Tabus, “Depth image lossless compression using mixtures of local predictors inside variability constrained regions,” in *Int. Symposium on Communications, Control, and Signal Precessing*, Rome, May 2012.

[2] Y. Morvan, D. Farin, and P. H. N. de With, “Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images,” in *Proc. IEEE Int. Conf. Image Processing*.

[3] W.-S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, “Depth map distortion analysis for view rendering and depth coding,” in *Proc. 16th IEEE Int Image Processing*, 2009, pp. 721–724.

[4] C. Dal Mutto, P. Zanuttigh, and G. M. Cortelazzo, “Scene segmentation by color and depth information and its application,” in *STreaming Day*, Udine, Italy, Sept. 2010.

[5] P. Zanuttigh and G.M. Cortelazzo, “Compression of depth information for 3D rendering,” in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video*, Potsdam, Germany, May 2009, pp. 1–4.

[6] C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, “High-quality video view interpolation using a layered representation,” in *ACM SIGGRAPH and ACM Trans. on Graphics*, Los Angeles, 2004, pp. 600–608.

[7] M. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS,” *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, 2000.

[8] I. Tabus and S. Sarbu, “Optimal structure of memory models for lossless compression of binary image contours,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Prague, May 2011, pp. 809–812.