MDL SEGMENTATION AND LOSSLESS COMPRESSION OF DEPTH IMAGES

Ionuţ Şchiopu and Ioan Tăbuş

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 proposes methods for lossless compression of depth images, where the intermediate stage of image segmentation produces a minimum description length (MDL) segmentation, which is realized so that the overall description of the regions obtained and of the residuals obtained over each region is minimized. The existing methods for image segmentation based on minimum description length normally consider prediction over the regions by using planar or second order models. Differently, in here non-linear prediction is utilized in each region, making it possible to achieve compression ratios much better than the current standards for lossless compression. The standard lossless compression methods, which were designed for natural images, prove to not be the most effective way to encode depth images, because depth images are more redundant and have different regularities than natural images. The newly proposed technique reduces the size of the compressed files in average to 55% of the standard JPEG-LS for a wide range of depth image material.

1. BACKGROUND TO MDL SEGMENTATION

Finding the segmentation of an image based on the minimum description length was first considered in [2], where the optimization of the MDL criterion was done through a costly continuation method. A much faster and more intuitive optimization process was introduced later by using region merging segmentation driven by MDL criteria in [1][3], where multichannel images were also considered. The lossless description of the image is realized by encoding four items: 1) the contours of the regions which make up the segmentation description; 2) the parameters of the polynomial models over each region; 3) the parameters of one multivariate (in the case of multichannel image) Gaussian model for the residuals over each region; and 4) the residuals over each region, encoded based on the statistical models specified at item 3). This representation of the image has the significance of decomposing the original image in a cartoon like image where each region has a given color, plus a residual image, which will include texture and random noise as well. The fact that the residual image is modeled over each region only as a multivariate Gaussian will leave unexploited a lot of redundancies present in regular textures or other local regularities inside regions, and as a result the overall compression of the above scheme is inferior to most lossless compression schemes. A similar approach was recently used for the *lossy* compression of depth images in [4]. We present in this paper a different scheme, where the modeling of the regions is done by more efficient predictive tools and as a consequence the compression obtained with our scheme is competitive and even exceeding by a large margin the performance of the best lossless coders. At a conceptual level, finding the segmentation which minimizes the description length in our scheme is really the minimum description length segmentation of the dept image. In the following we give a brief account of the algorithmic solutions involved and for a detailed description we refer to [5].

2. A NEW METHOD FOR MDL SEGMENTATION

We start by defining the costs for each region. A region is defined by the set Ω of pixel coordinates (x, y) which belong to that region. The image graylevel at pixel (x, y) is $I(x, y) \in \{0, 2^B - 1\}$, where the number of depth-planes is usually $B = 8$. We want to predict the depth $I(x_t, y_t)$ at a current pixel (x_t, y_t) by using the values $I(x_i, y_i)$ at the pixels (x_i, y_i) from a causal neighborhood $\mathcal{N}(x_t, y_t)$ of the pixel (x_t, y_t). We note that this predictive principle is used in all competitive lossless image compression schemes. The shape and size of the neighborhood selected in different compression schemes varies quite much, the most simple shapes being those including only the west, north, north-west pixels. Here we adopt two scanning orders for the pixels in a neighborhood: horizontal, i.e. along the rows of the image, or vertical, i.e. along the columns of the image, and use over each region that scanning which gives the best results. The causal neighborhood for the horizontal scanning will have four pixels: W,NW,N, and NE, while the neighbors for the vertical scanning will be: W,NW,N, and SW.

2.1. Prediction

We want to define a suitable segmentation of the image, where any region Ω contains pixels with identical, or similar graylevel values, and the pixels outside Ω may have very different graylevel values. However a region may contain also pixels having the property that their values are well predictable using a given prediction method and a given prediction mask, while the pixels outside the region are not anymore predictable. For this reason, the causal neighborhood $\mathcal{N}(x_t, y_t)$ is restricted only to pixels belonging to Ω, and all pixels from outside Ω are excluded. Close to the borders of the regions, the neighborhood $\mathcal{N}(x_t, y_t)$ will not have exactly 4 pixels, it may have 3,2 ,or 1, or even zero pixels. We need to define a suitable predictor function and we consider here $\hat{I}(x_t, y_t) = median\{I(x_i, y_i)|(x_i, y_i) \in \mathcal{N}(x_t, y_t) \cap \Omega\}$, having the support with a varying number of pixels. The pixels having the intersection $\mathcal{N}(x_t, y_t) \cap \Omega$ empty are collected in a set Δ and we use for them the prediction $\hat{I}(x_t, y_t) = mean\{I(x_i, y_i)|(x_i, y_i) \in \Delta\}$ which is encoded and transmitted for each region as a header.

2.2. Encoding of prediction residuals

We fix a scanning order of the pixels in the region Ω having N pixels, and denote by (x_t, y_t) the coordinates at location t in the region, with $t = 1, \ldots, N$. We define the residuals $\epsilon(x_t, y_t) = I(x_t, y_t) - \hat{I}(x_t, y_t)$ for all pixels $(x_t, y_t) \in \Omega$ and denote the minimum and maximum residuals by m and M. Both m and M will be encoded before encoding the region, with m encoded uniformly in $\{-2^B + 1, 2^B - 1\}$, and $m_s = M - m$ encoded after that uniformly in $\{0, 2^B - m - 1\}$. Since m and M will be available also at decoder, we are going to encode $\varepsilon(x_t, y_t) = \epsilon(x_t, y_t) - m \in \{0, M - m\}$. In case $m = M$ there is no need to encode the residual, for all pixels in the region we have $I(x_t, y_t) = m = M$.

In the case $m \neq M$ we need to encode the residuals $\varepsilon(x_t, y_t)$, which will be done by using the adaptive distribution collected while encoding the residuals, in the agreed scanning order. Let denote the counts $n_t(i), i = 0, \ldots, M - m$, which tell how many times $\varepsilon(x_j, y_j)$ was equal to i for all $j \leq t$. The distribution of the residuals is tracked and used adaptively, so that at the current pixel $(x_t, y_t))$ both encoder and decoder have available the set of counts $n_{t-1}(i), i = 0, \ldots, M - m$, which tell the frequency of the symbols observed up to and including $t - 1$'th pixel. We can thus encode $\varepsilon(x_t, y_t)$ using $\mathcal{L}_t = -\log_2(n_{t-1}(\varepsilon(x_t, y_t)) / \sum_i n_{t-1}(i))$. The overall codelength will add all such elementary codelengths over one region.

2.3. Encoding of region contours

The overall segmentation is formed of the contours separating the regions. There are a number of strategies for encoding these contours and their starting points. The order in which boundaries are transmitted will affect the number of starting points (which we also call anchors) and of ending points. We tested a number of heuristics and choose the one offering the lowest cost. We encode the vertex chains using the 3OT chain-code representation encoded with adaptive-order Markov models [6].

2.4. The overall segmentation method

With the costs as defined above, a MDL segmentation can be obtained by starting from an initial over-segmentation obtained e.g. as in [3] and then performing a very laborious sequential merging process, where two neighbor regions are merged if the overall code-length is better after merging. In order to accelerate the segmentation process we found a much faster procedure, probably suboptimal but still extremely efficient, summarized in the following: take as initial over-segmentation of the image the split into regions depending on a variable called *Threshold*, defined as follows. A pixel will belong to a given region if the absolute value of the difference between the pixel depth and the depth of one of its 4-connectivity neighbors is smaller than *Threshold*. At the first step we find the regions with *Threshold =1* (they are constant regions) and keep the regions which are large enough as they are (no further merging is attempted). The smaller regions are collected together and a new split process takes place, allowing this time more variability inside regions, by increasing the *Threshold* to 2. Again, the regions which are large enough are kept unchanged. The process continues with *Thresholds* 3 and 4, and only the remaining small regions are checked if their merging produces improvements in the overall codelength. Finally, the encoding of remaining very small regions of up to 4 pixels is performed in a number of specific ways. The detailed processing is illustrated in Figure 5 and we send for details to [5]. The overall encoding strategy is summarized in Figure 1.

2.5. Experimental results

We illustrate the segmentation algorithm by the segmentation in Figure 4 for the depth image presented in Figure 3 (for completeness we also show the corresponding color image in Figure 2). The resulting MDL segmentation relies on more regions than a human will tend to associate to the image if he would intend to get only a sketchy cartoon of the image. However, the complete lossless representation of the image require such an over-segmented image in order to obtain really a minimum description of the whole depth image.

For illustrating the lossless compression performance we present results for a set of images from [8]. The comparison with the standard JPEG-LS [7] compressor (using the implementation provided in [9]) and with the PNG compressor (PNG being the format normally used for storing depth image in the public databases) is illustrated in Table 1, where compression factor (CF) is defined as compressed size over initial size, showing a very good performance of our encoder. All results are double checked for perfect reconstruction of the original file after decoding.

More comparisons in [5] for about 200 frames of two depth image sequences show that indeed the lossless compression performance of the presented scheme overpasses significantly that of commonly used standard lossless image compression methods.

3. REFERENCES

[1] T. Kanungo, B. Dom, W. Niblack, and D. Steele, “A fast algorithm for MDL-based multi-band image segmentation,” in *IEEE Conference on Computer Vi-*

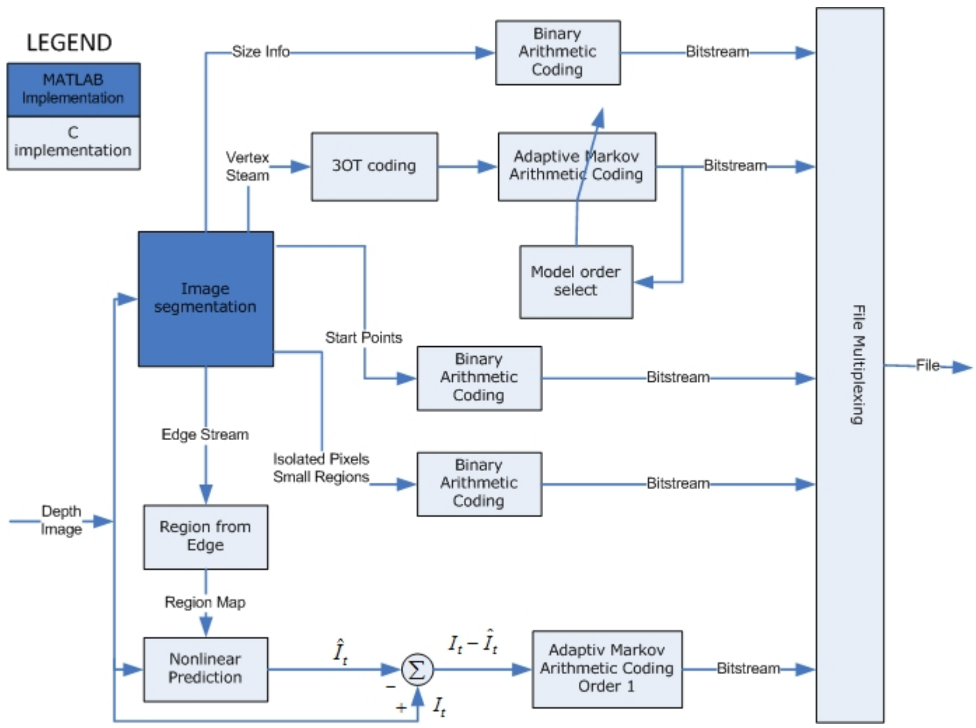

Figure 1: The encoding strategy for depth image compression.

Image Name	Initial size (bits)	PNG size (bits)	CharLS size (bits)	Our new meth. result (bits)	CF PNG [%]	CF CharLS [%]	CF of our new meth. [%]
Art	1370480	194736	212704	**107080**	14.20	15.52	**7.81**
Books	1370480	159560	152720	**104208**	11.64	11.14	**7.60**
Dools	1370480	272136	232496	**177840**	19.85	16.96	**12.97**
Laundry	1323120	168080	157824	**102640**	12.70	11.92	**7.75**
Moebius	1370480	193848	170272	**109424**	14.14	12.42	**7.98**
Reindeer	1323120	187112	174224	**113768**	14.14	13.16	**8.60**
Lampshade	11544000	611288	300561	**245768**	5.29	2.60	**2.13**

Table 1: Results and compression factors (CF) for the set of different depth images. With bold text are presented the best results.

sion and Pattern Recognition, Seattle, pp. 609–616, June 1994.

[2] Y.G. Leclerc, "Constructing simple stable description for image partitioning," *International Jounal of Computer Vision*, vol. 3, pp. 73–102, 1989.

[3] Q. Luo and T. M. Khoshgoftaar, "Unsupervised multiscale color image segmentation based on MDL principle," *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2755–2761, 2006.

[4] S. Milani and G. Calvagno, "A Depth Image Coder Based on Progressive Silhouettes", *IEEE Signal Processing Letters*, vol. 17, issue 8, pages 711-714, 2010.

[5] I. Şchiopu, "Segmentation and compression of depth images", *M.Sc. thesis*, Tampere University of Technology, May 2011.

[6] I. Tăbuş and S. Sârbu, "Optimal structure of memory models for lossless compression of binary image contours", IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2011, May 22-27, Prague, 2011.

[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, issue 8, pages 1309-1324, 2000.

[8] Repository *vision.middlebury.edu*: Stereo Datasets [Online]. Available: *http://vision.middlebury.edu/stereo*.

[9] Source files of CharLS arhive [Online]. Available: *http://charls.codeplex.com/* Version 1.0.

Figure 2: *Reindeer*: color image.

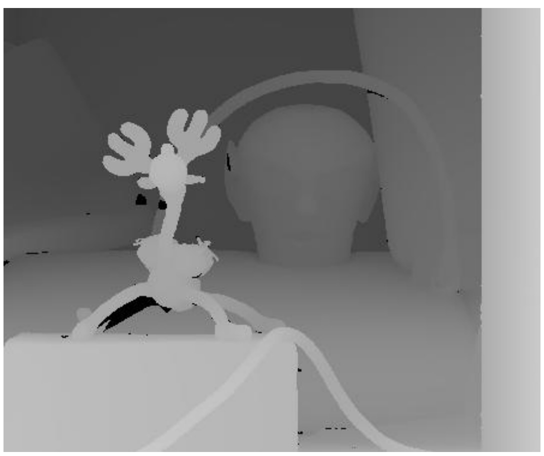

Figure 3: *Reindeer*: depth image.

Figure 4: Segmentation of image *Reindeer*

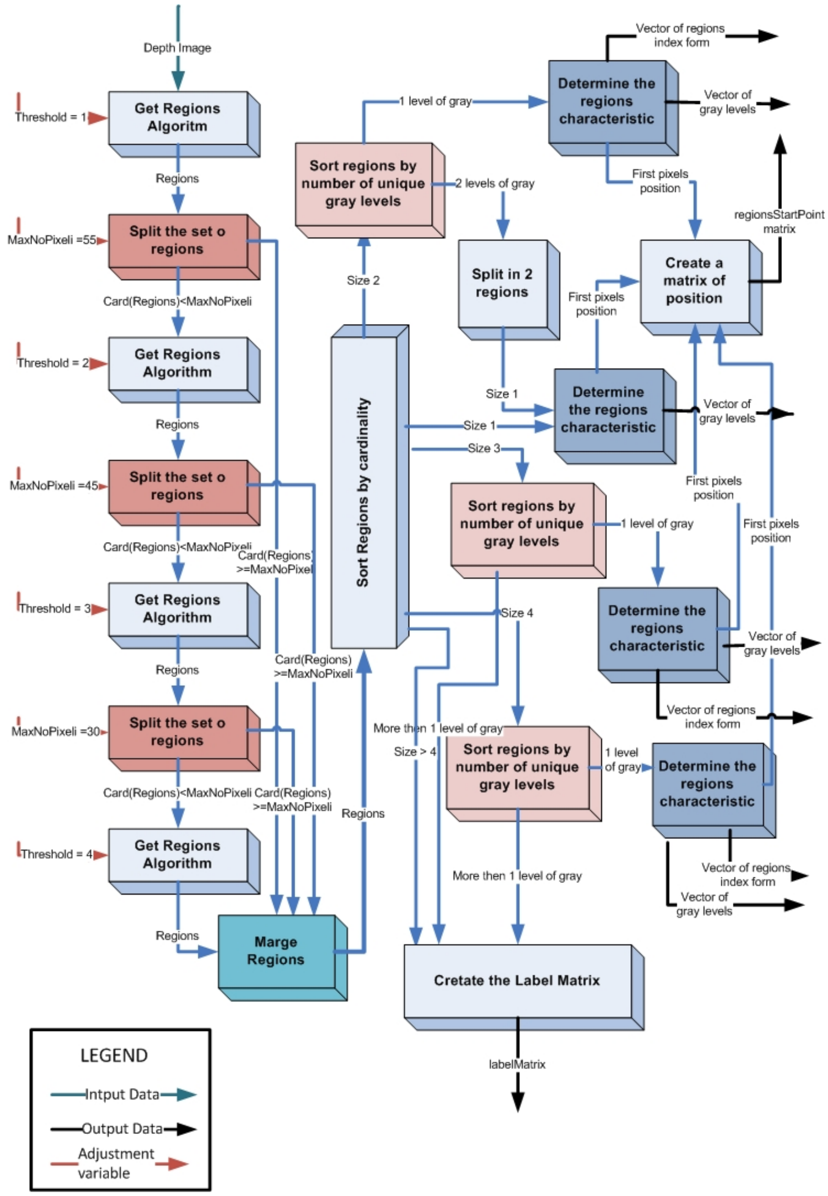

Figure 5: Image segmentation diagram.