WAVELET-BASED COMPRESSION OF SEGMENTED IMAGES

Radoslav Vargic

Slovak University of Technology in Bratislava, Faculty of Electrical Engineering and Information technology, Ilkovičova 3, 812 19 Bratislava, Slovak Republic, vargic@ktl.elf.stuba.sk

Abstract: *In this article we present an approach to compression of segmented images using wavelet transform The original image is segmented using an unsupervised regions segmentation method.. The segments are approximated using polygons to minimise the representation. For texture approximation we use iterated wavelet orthogonal transform. We develop a new scheme for approximation and coding which optimises the results in rate-distortions sense. Proposed scheme gives comparative results to [1] but with significant speed-up of computation.*

Key words: *wavelet transform, segment representation, region coding*

1. INTRODUCTION

Region and object oriented methods allow better adaptation to the local image characteristics. There are needed algorithms that can deal with arbitrarily shaped image regions, called segments, instead on rectangular blocks. There were proposed more application schemes, e.g. using DCT [2] or complete image coding process [1] using various orthogonal transforms. In this article, there are found region boundaries and consequently approximated with polygonal lines similar to approach in [1]. The texture inside each region is approximated using discrete wavelet transform (DWT) defined on the rectangle circumscribing the given image segment. Each segment is coded using SPIHT [4] algorithm, with bit budget proportional to segment size. Transformation is iterated to give the best approximation on given initial approximation and bit budget. The results are compared with ones from [1] and non-segmented approach.

2. IMAGE SEGMENTATION, APPROXIMATIONS AND CODING OF THE REGIONS

To partition the image we used an unsupervised segmentation method for colour –texture regions used in [1]. The segments are found in a few steps. The result is an image, where each region is defined by its unique gray value. Region boundaries are found using 8-directional algorithm based on LML (left-most-looking) as in [1]. Boundaries of the segments are then approximated with polygons. Whole data stream consisting of all encoded boundaries is encoded using the modified Huffman code with DPCM. The whole process can be further extended using triangularisation methods [5] to produce smaller regions, more suitable for some texture coding methods[2]. Example of segmented image and map of polygonalised segments is depicted in Fig.1.

a) **b)** **c)**

Fig. 1. Sample input image (baboon): a) segmented into 12 segments b) map of polygonal approximations for 12 segments c) polygonal map for 5 segments

3. TEXTURE APPROXIMATION AND CODING

Main topic addressed in this article is the texture approximation and coding inside of each segment. The first step is to choose the suitable approximation, then to code it.

Initially we followed the approximation scheme sketched in [2], which is used with orthogonal transforms in [1] and with wavelet transforms in [3]. There the segment approximation is performed iteratively, adding only one contribution to optimally chosen basis function to assure the best successive approximation property („matching pursuit“ approach) inside of the region. When the approximation is sufficient, the coefficients are coded, causing additional quantisation error. This approach has two main drawbacks:

1) The choosing and adding contributions for individual basis functions in bigger segments is very time expensive.
2) There is no check if the coder will use our carefully chosen basis functions, i.e. no test if the contributions are optimal in the sense of additional costs of coding

To overcome drawback 1) can be used e.g. triangularisation of regions [5]. To handle the drawback 2, one should iterate until the best approximation is reached and then let the coder to code as much information as possible [1]. There is desired to do on-the-fly rate-distortion optimisation when constructing the approximation, i.e. adding differences to the individual basis functions.

3.1. PROPOSED ALGORITHM

The proposed algorithm uses DWT, which is used on rectangle $\boldsymbol{L}$ with dimensions of power two circumscribing the segment $\boldsymbol{A}$. There are infinitely many possibilities how to represent segment $\boldsymbol{A}$ in some basis on rectangle L. When approximation is sufficient, then the goal is to find the best approximation of segment $\boldsymbol{A}$ using given transform and coder on given bit rate. In proposed algorithm we try to address also this topic. Our algorithm does not find global optimal representation; only given initial extrapolation finds better approximation with respect to given coder and bit budget. The DWT does need to be orthogonal as in [1][3], bi-orthogonal condition is for iteration sufficient. The main principle is to let the segment $\boldsymbol{A}$ itself to create its extension on L which is needed for most effective coding of segment $\boldsymbol{A}$. The appropriate extension is needed to eliminate the effect of discontinuity at segment boundaries.

In proposed algorithm the extension arises in iteration process as consequence of limited bit budget for coding. Limited budget causes "blurring" on image boundaries, which is re-used in next iteration. Clearly if we have unlimited bit budget, there arises no extension and iteration to create the best extension has no sense. Because of linearity of the transform, this iteration approach is similar to approach used in [2] changed not only to update individual coefficient but all coefficients at once in one iteration step. The iteration process is depicted on Fig. 2. Cases A and B are equivalent (the only difference is that A is processed without mean component) and represent the initial extrapolations with mean value. Case A can be viewed as first modelling the image with its average and coding only the difference against this. Case C represents bad initial approximation, which is similar with changed approach [2] as stated above. Case D represents custom initial extrapolation.

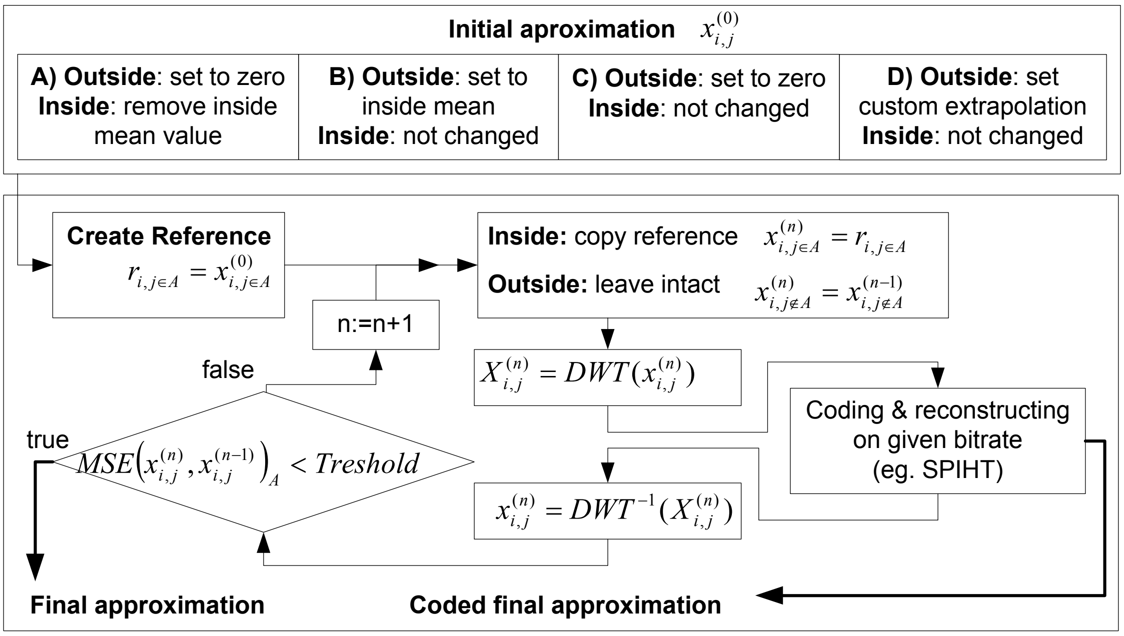

Fig. 2. Iterative algorithm for coding of 1 segment using DWT and SPIHT

4. COMPRESSION RESULTS

Given initial approximation our iterative DWT approach enhances the MSE of final approximation about 1% when using mean extrapolation (Case A or B) and about 10% when using zero extrapolation (case C). For comparison with other methods we use the Case A coupled with SPIHT coding of coefficients. We refer to this approach as WSEG. We stopped the iteration when MSE difference between successive image approximations was less than 0.001. This was typically reached in 2-7 iterations. Example of visual impact of iteration one can see in Fig. 5. We compared our compression results to non-segmented compression of image using SPIHT algorithm and to best results in [1] obtained using iterated DCT II. The compression results comparison for Baboon image with 5 segments is graphed in Fig 4.

a) b) c)

Fig. 3. Baboon image compressed at 0.1 bpp: a) not segmented (DWT with BattleLemaire(BL) filters +SPIHT) b) WSEG with 5 segments (DWT with BL filters +SPIHT) c) Iterative DCTII from [1] with 5 segments

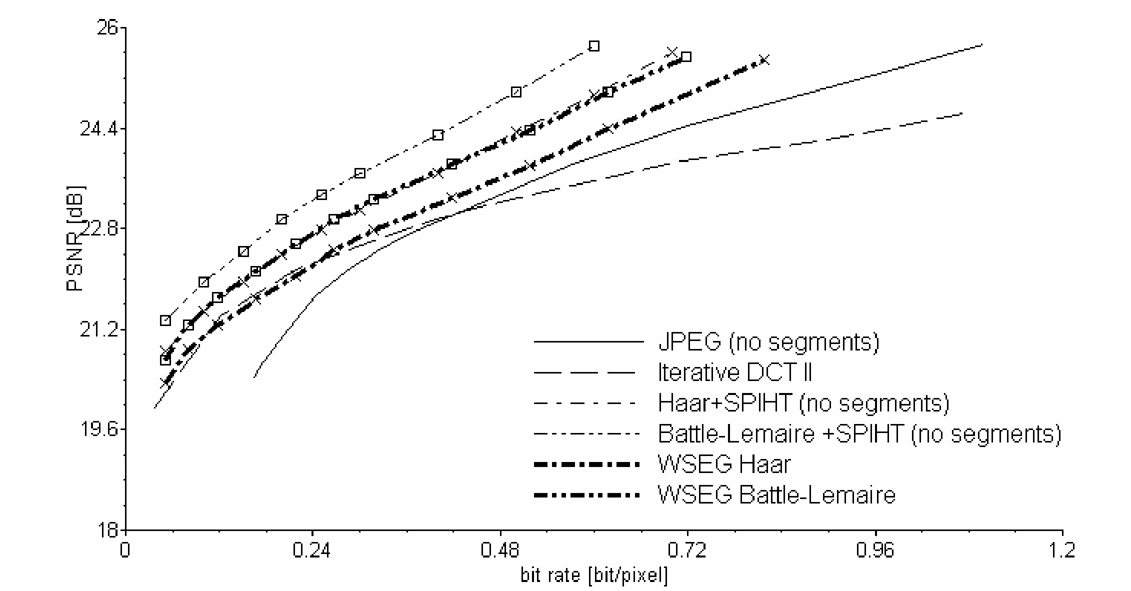

Fig. 4. Comparison if compression results for segmented “baboon” image with 5 segments

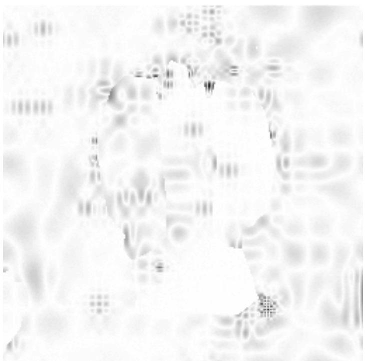

Fig. 5. Image of absolute differences between initial and iterated segment approximations in baboon with 5 segments. Used Battle-Lemaire DWT on 0.1 bpp, 7 iterations and extrapolation with mean. Small values are depicted as white.

4. CONCLUSION

As one can see our method is not so effective in PSNR sense as non-segmented version, but still more effective and significantly faster as segmented approach [1]. The advantage (when properly segmented) over non-segmented approach is clearly visible segment boundaries and no error spreading between segments (see baboon' nose in Fig.3). However when the number of segments grows (say 10 or above), there is significant loss in coding effectively in our method, because too many bits are used to code positioning information in multiple SPIHT hierarchical trees. So we propose to use proposed method with SPIHT algorithm only for small number of segments. When more segments should be involved, the coding infectivity can be avoided using classical "baseline" spectral coding approach as in [6].

REFERENCES

[1] Karlubíková, T. et al., New Scheme for Region Approximation and Coding with Shape Independent Transform, Proc. Photogrammetric Computer Vision (PCV'02). Graz, Austria, 2002, ISPRS Commision III, Vol. 34, Part 3B, ISSN 1682-1750, pp. 214-217.

[2] Kaup,A., Aach,T., 1998. Coding of Segmented Images Using Shape-Independent Basis Functions. IEEE Trans. on Image Processing, Vol.7, No.7, pp. 937-947.

[3] Y. Liu, S. Ranganath, X. Zhou, "Wavelet-based Image Segment Representation", Electronics Letters, Vol. 38, No. 19, pp. 1091-1092, 2002.

[4] Said, A., Pearlman, W., A. A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," by Amir Said and William A. Pearlman, *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 6, pp. 243-250, June 1996

[5] Polec,J., Kolingerová, I., Partyk, M., Hybrid Approach with Polygonization and Triangulation for Transform Coding, proc. East-West-Vision 2002 (EWV'02). Graz, Austria, 2002, ISBN 3-85403-163-7, pp. 219-220.

[6] Comaniciu, D., Meer, P., Image Segmentation and residual Coding for Very Low Bit Rate Compression, http://www.caip.rutgers.edu/riul/research/papers/pdf/compression.pdf