

A Generalization of Quad-Trees Applied to Lossy Image Coding

Rade Kutil

University of Salzburg, Department of Computer Sciences
 Jakob Haringer-Str. 2, 5020 Salzburg, Austria
 Email: rkutil@cosy.sbg.ac.at

Abstract—Quad-trees are restricted to combined horizontal and vertical decomposition of tiles. If this restriction is dropped, anisotropic rectangular tiles result which can be arranged in a previously developed graph structure called “bush”. Although this graph requires more bits to be encoded, the reduction of the number of tiles by a factor of about two more than compensates this, especially when there is a big payload per tile such as polynomial or DCT approximations of image content. Therefore, together with an existing efficient coding scheme for this graph, a lossy coding scheme based on tile-wise rate-distortion optimized quantization of low-frequency DCT coefficients is developed. It is able to compete with JPEG2000, especially for images with smooth regions.

I. INTRODUCTION

Quad-trees have long been used in general image coding [1], for bi-level images [2], and video coding [3]. In quad-tree coding, square tiles are recursively decomposed into four square sub-tiles. The process stops for sufficiently uniform tiles, for which the color payload is encoded. The tree structure also has to be encoded. However, the advantage of quad-trees is that they can be encoded efficiently by only one bit per tile that indicates if it is split or not.

For bi-level or indexed color images, the payload is the pixel color. For natural images, image segments are approximated by planar [4] or polynomial [5], [6] functions. Generalized tilings with arbitrarily oriented linear splitting are used here, though, to achieve a more accurate approximation of region borders. However, this leads to increased bit budgets for encoding of the segmentation structure, so block merge algorithms [7] or combinations with quad-trees [8] have been developed.

These schemes can be applied to DCT [9] and wavelet [10] coding, as well as motion estimation [11], [12], [13] in video coding. Common to these applications is the amount of data to be encoded per tile, which is larger than for bi-level images, where JBIG2 [14] or chain codes [15] are far superior. Another reason to use tree structures to encode image data is the ability to arbitrarily select spatial details, as needed in terrain visualization [16] and display of geospatial data [17]. Also, spatial databases use quad-trees [18] in a similar way.

This work exploits a generalization of quad-trees. Tiles may be split anisotropically in horizontal or vertical dimension, which may produce highly non-square tiles. In this way, many shapes can be represented with only about half the number of tiles, as can be seen in Figure 1. The number of these decompositions, however, was shown to be much higher than

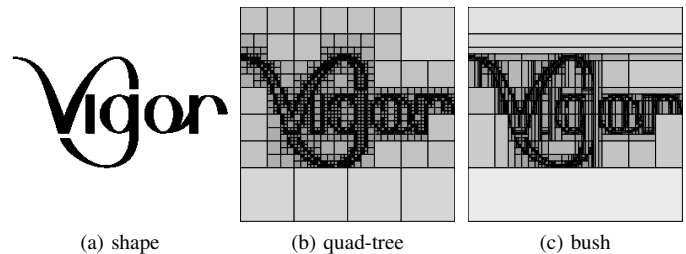


Fig. 1. A shape and its decomposition into 3514 quad-tree tiles and 1861 bush tiles.

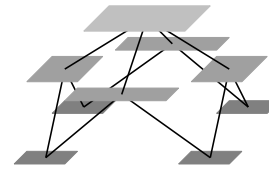


Fig. 2. A full bush of anisotropic tiles

that of quad-trees [19], [20]. Moreover, the representation as a binary tree of horizontal or vertical splits is not unique, and, therefore, causes redundancy and inefficiency in coding. However, if the graph structure is expanded to incorporate all possible decomposition trees with the same set of leaf nodes, uniqueness is achieved. See Figure 2. In [21], [22], [20], such a graph, called “bush”, together with an efficient redundancy-free coding algorithm has been developed. Redundancy-free means that there is only one representation for each set of tilings and, when encoded, no encoded symbol can be deduced from other parts of encoded data. An efficient coding algorithm is important in our context because the amount of data necessary to represent the tiling structure is not negligible compared to the coding of color payload.

After anisotropic tilings have been explored in the lossless case in [23], this work presents a lossy image compression algorithm in order to demonstrate the benefits of bush tilings on general images. It applies successive bitplane coding of a subset of low-frequency DCT coefficients, which are calculated for each tile. These DCT coefficients are considered as a tile-wise approximation in the sense of piece-wise planar or polynomial approximations as in [5], [6]. The bit-rate per tile is controlled by the number of bitplane passes performed. The optimal number of passes for each tile is determined by rate-

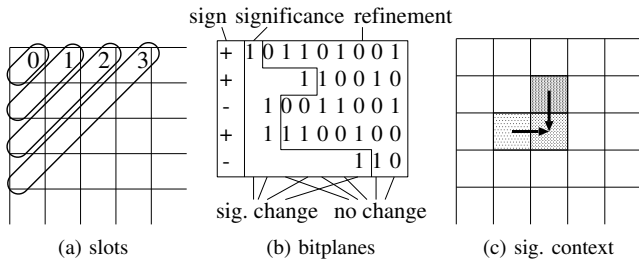


Fig. 3. coding of DCT coefficients

distortion optimization. The best tiling is also found in terms of this rate-distortion optimization by minimizing the sum of rate-distortion values, which leads to a variant of the optimal tiling algorithm of the lossless case as in [23]. The schemes are compared to JPEG2000.

II. CODING ALGORITHM

In order to use tilings in lossy image coding, we need a reasonable coding algorithm that is applicable to rectangular tiles. Each tile should represent a good local image approximation. Refinements on the approximation shall not be achieved by a more detailed description of the contents of a tile, but by splitting a tile into smaller ones that can be approximated more easily.

Therefore, we perform a discrete two-dimensional cosine transform (DCT) on the tiles and select a subset of the coefficients as the approximation of the tile's content. The subset chosen is, of course, a set of low-frequency coefficients. It is organized as a number of *slots*. Each slot groups coefficients with horizontal frequency index i and vertical index j so that $i + j$ is the slot's index. See Figure 3 (a). Slot 0 contains the so-called DC coefficient. When n slots are used, then $n(n + 1)/2$ coefficients are to be encoded, all others are neglected. If other coefficients contain too much energy, so that the approximation error is too big, then the tile has to be split. The choice of DCT is, of course, motivated by JPEG, but also by [9] which uses DCT on quad-tree tilings. The shape of the slots is motivated by the zig-zag scan order of JPEG.

To encode the DCT coefficients efficiently, we apply a bitplane approach, as is usual in modern compression schemes [24], [25], [26]. Beginning with a maximum threshold, the most significant bits of the DCT coefficients are encoded so that the quantization error of each coefficient is smaller than the threshold. The quantization is then refined in subsequent passes with the threshold being divided by two in each pass. DC coefficients tend to dominate the energy in the tile's DCT domain. This problem is somewhat relieved by simply subtracting the average of the whole image from the image, which has to be encoded in the beginning.

After each pass we get a certain total bit-rate R and a total distortion D of the tile, i.e. the sum of squared approximation errors. The distortion can be calculated in the DCT domain because of the orthogonal nature of the DCT. The number of passes must be chosen for each tile so that the sum of rates

and distortions of all tiles is an optimal compromise, which is done with rate-distortion optimization. A rate-distortion slope λ is chosen for the whole image, and in each tile the point on the rate-distortion curve with the minimum RD-value $D + \lambda R$ is selected. This has been proven to produce the minimum distortion for the according total bit-rate [24], [25]. The total bit-rate can be adjusted by the choice of λ . The optimal number of passes has to be encoded for each tile.

However, the optimal tiling structure is not independent of the choice of rate and distortion, i.e. the choice of the RD-slope λ . A tile might be approximated better with a reduced bit-rate if it is split into sub-tiles. Therefore, we add the RD-values $D + \lambda R$ of the sub-tiles and compare the sum to the RD-value of the parent tile. To be more precise, not the RD-values of the sub-tiles themselves but the optimal values after the splitting decision for the sub-tiles are considered here, which leads to a recursive algorithm. If the sum is smaller, then the tile is split into four child sub-tiles in the case of quad-tree tiling. In the case of anisotropic bush tilings, splitting can be done in two possible dimensions. The dimension with the smaller sum of two RD-values is selected. This leads to a tiling algorithm equivalent to the one in [23], where RD-values are added and minimized instead of just tile counts.

Arithmetic coding is used to encode significance bits, refinement bits and sign bits of the DCT coefficients. See Figure 3 (b). A coefficient is considered significant if it is larger than the current bitplane threshold. If a coefficient becomes significant, then its sign bit has to be encoded as well. The coefficient is then approximated by 1.5 times the threshold at the decoder. The tile's distortion has to be reduced accordingly. Coefficients that have become significant in previous passes are updated by refinement bits.

Refinement and sign bits have a 50% percent probability for 1 and 0. Significance bits, however, correlate with those of neighboring coefficients. Therefore, they are classified into two contexts depending on whether the left or upper neighbor, i.e. $(i - 1, j)$ and $(i, j - 1)$, was significant in a previous pass. See Figure 3 (c). When those two neighbors are not significant, then there is a chance of only 10% or lower that the coefficient becomes significant. Otherwise, the probability is higher, i.e. about 50% for more than 3 slots. Additionally, a pass with no changes in coefficient significance is abbreviated by a single bit for the whole pass. This bit has a probability of $2/3$ for no significance change. These probabilities are used in a fixed way in the arithmetic coder, no adaptivity is applied. This has the advantage of better computational performance and, more important, the rate-distortion analysis is exact because bit statistics of neighboring tiles have no influence on the bit-rate.

This coding scheme is supposed to be suitable for smooth regions with sharp borders, which corresponds to shape coding in lossless image coding. Images like that occur as technical drawings, diagrams and cartoons. It is assumed that sharp borders can be better approximated by tile borders or small tiles. Anisotropic tiles should be able to adapt to borders with a lower number of tiles, thus saving a lot of bit-rate.

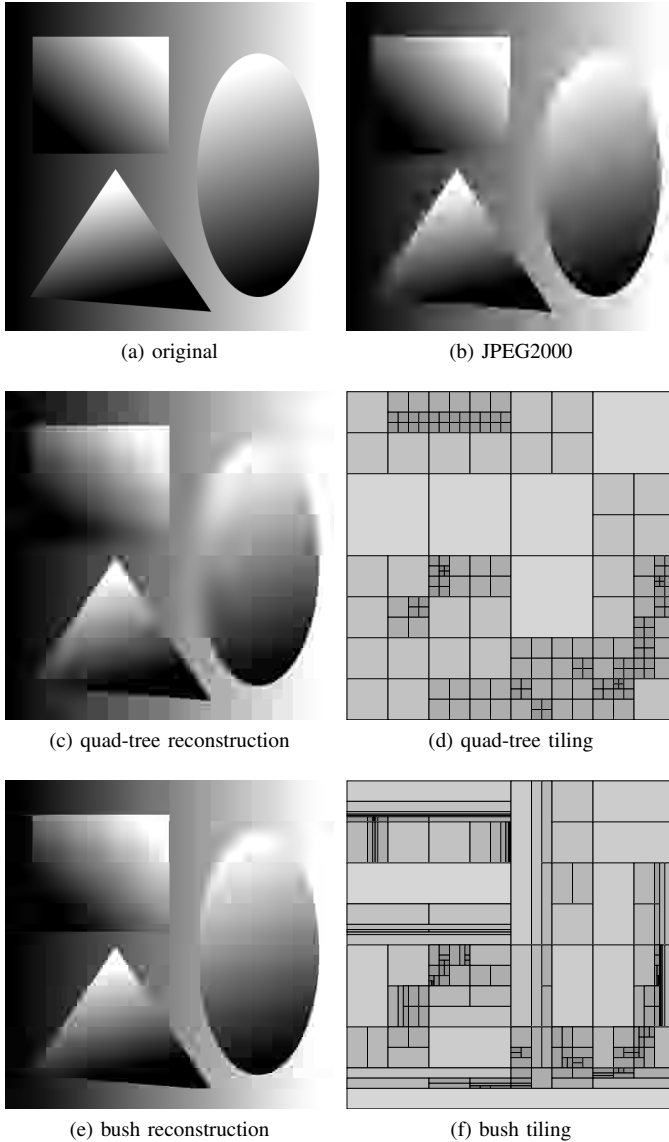


Fig. 4. Compression of the gradient-shape image at 0.02 bits per pixel with 5 DCT slots

As the “color payload” in this case consists of encoded DCT coefficients and is large compared to the simple color indices of the lossless case, anisotropic tilings should show advantages. All this is tested in the next section.

III. EXPERIMENTAL RESULTS

The lossy coding algorithm is tested on an artificial image with a square shape that is horizontally and vertically aligned borders, a triangular shape with angular borders, and an elliptic shape. See Figure 4 (a). The shapes are filled with smooth gradients. Therefore, we will call this image the *gradient-shape* image. It is supposed to be suitable for the proposed tiling-oriented coding algorithms. Results for the more natural Lena image will also be mentioned later on.

In Figure 4 (b) one can see that JPEG2000 produces very

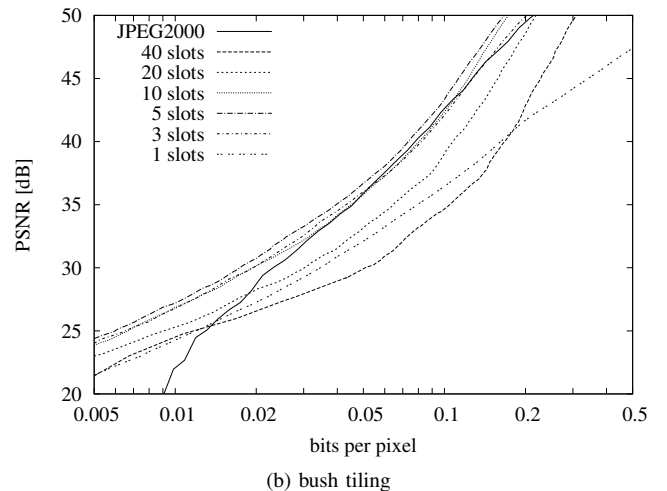
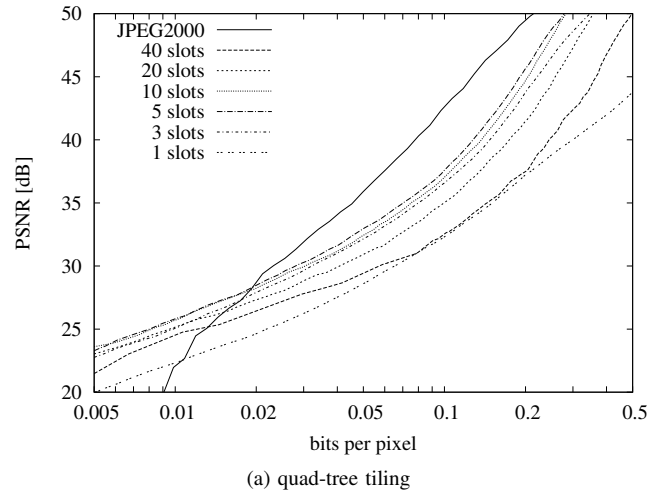


Fig. 5. Image quality depending on bit-rate for the gradient-shape image

blurred edges for a low bit-rate of 0.02 bits per pixel. The quad-tree coder is not able to improve this because creating more small tiles at edges would also increase the bit consumption for neighboring small tiles where it is not needed. However, bush tilings again improve the adaptivity of the tiling to image content significantly. This produces a much clearer representation of shape borders with less ringing effects.

The fact that quad-tree tilings are not suitable for efficient coding of the shapes is even more distinct in Figure 5 (a). JPEG2000 has an up to 5 dB better PSNR than the quad-tree coder for all feasible bit-rates. Only for very low bit-rates, JPEG2000 has worse results probably due to the higher amount of header information. Using 5 DCT slots seems to be the best choice.

There is a big difference, however, between quad-tree and bush performance, as can be seen in Figure 5 (b). The bush tiling scheme is able to outperform JPEG2000 slightly for mid-range qualities, and significantly for low and high bit-rates. The best number of DCT slots is 5, just as for the quad-tree tiling. In summary, anisotropic bush tilings are able to improve

the rate-distortion performance by about 5 dB against quad-tree tilings.

For the Lena image, the results are similar but less dramatic. The bush-tiling coder cannot beat JPEG2000, but is less than 0.5 dB worse, whereas the quad-tree coder is about 1.5 dB worse than JPEG2000. A higher number of DCT slots, 10, is the best choice for this image. The reason is that less smooth tile content requires higher frequency coefficients.

Other natural images (Barbara, Goldhill, Peppers, Baboon) show about the same behavior as the Lena image, i.e. 0.5 dB below JPEG2000 for bush tilings, and 1 to 2 dB for quad-trees. This shows that the codec is more suitable for technical drawings, presentations, or other artificial images. Interestingly, a cartoon image, while also not beating JPEG2000 in terms of PSNR, has a much better visual quality.

The share of bits in the bitstream for encoding the quad-trees is 0.5 to 1% for Lena and 2 to 3% for the gradient shape image. In the bush case we need 6 to 8% for Lena and 11 to 13% for gradient shape. This shows that, although the bush structure is more complex, it is able to more than compensate that with bit savings in the encoding of the tile's content. The ratio is usually lower for higher image quality.

IV. CONCLUSIONS

Anisotropic tilings are able to represent a shape with only half the number of tiles compared to quad-tree tilings if a new algorithm for optimal tiling is applied. However, a bigger part of the bit-rate has to be devoted to encoding the tiling, although this can be done efficiently with a generalization of trees, called "bush". Nevertheless, the reduced number of tiles reduces the bit-rate for the payload, i.e. the tile color information. A lossy image compression scheme proposed in this work is based on DCT approximation. It distributes bit-rates among tiles by applying rate-distortion optimization. The optimal tiling is found by an algorithm that minimizes the total rate and distortion. Results show that bush tilings gain up to 5 dB of image quality compared to quad-tree tilings. The scheme is able to compete with JPEG2000 and outperforms it for artificial images containing shapes of smooth content.

REFERENCES

- [1] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of images and video," *IEEE Transactions on Image Processing*, vol. 3, no. 3, pp. 327–331, May 1994.
- [2] M. Manohar, P. S. Rao, and S. S. Iyengar, "Template quadtrees for representing region and line data present in binary images," *Computer Vision, Graphics, and Image Processing*, vol. 51, no. 3, pp. 338–354, 1990.
- [3] M. Lightstone and S. K. Mitra, "Quadtree optimization for image and video coding," *Journal of VLSI Signal Processing*, vol. 17, pp. 215–224, 1997.
- [4] M. Sarkis and K. Diepold, "Content adaptive mesh representation of images using binary space partitions," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 1069–1079, May 2009.
- [5] R. Shukla, P. L. Dragotti, M. N. Do, and M. Vetterli, "Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 343–359, Mar. 2005.
- [6] H. Radha, M. Vetterli, and R. Leonardi, "Image compression using binary space partitioning trees," *IEEE Transactions on Image Processing*, vol. 5, no. 12, pp. 1610–1624, Dec. 1996.
- [7] C. S. Won, "A block-based MAP segmentation for image compressions," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 592–601, Sep. 1998.
- [8] A. A. Kassim, W. S. Lee, and D. Zonoobi, "Hierarchical segmentation-based image coding using hybrid quad-binary trees," *IEEE Transactions on Image Processing*, vol. 18, no. 6, pp. 1284–1291, Jun. 2009.
- [9] K. Lengwehasatit and A. Ortega, "Rate-complexity-distortion optimization for quadtree-based DCT coding," in *Proceedings of the IEEE International Conference on Image Processing, ICIP 2000*, vol. 3, Sep. 2000, pp. 821–824.
- [10] C. Y. Wang, S. J. Liao, and L. W. Chang, "Wavelet image coding using variable blocksize vector quantization with optimal quadtree segmentation," *Signal Processing: Image Communication*, vol. 15, no. 10, pp. 879–890, 2000.
- [11] V. Argyriou and T. Vlachos, "Quad-tree motion estimation in the frequency domain using gradient correlation," *IEEE Transactions on Multimedia*, vol. 9, no. 6, pp. 1147–1154, Oct. 2007.
- [12] J. Zhang, M. O. Ahmad, and M. N. S. Swamy, "Quadtree structured region-wise motion compensation for video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 5, pp. 808–822, Aug. 1999.
- [13] I. Rhee, G. R. Martin, S. Muthukrishnan, and R. A. Packwood, "Quadtree-structured variable-size block-matching motion estimation with minimal error," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 1, pp. 42–50, Feb. 2000.
- [14] F. Ono, W. Rucklidge, R. Arps, and C. Constantinescu, "JBIG2 – the ultimate bi-level image coding standard," in *Proceedings of the IEEE International Conference on Image Processing, ICIP 2000*, vol. 1, Sep. 2000, pp. 140–143.
- [15] H. Sánchez-Cruz, E. Bribiesca, and R. M. Rodríguez-Dagnino, "Efficiency of chain codes to represent binary objects," *Pattern Recognition*, vol. 40, no. 6, pp. 1660–1674, 2007.
- [16] K. Baumann, J. Döllner, K. Hinrichs, and O. Kersting, "A hybrid, hierarchical data structure for real-time terrain visualization," in *Proceedings of the Computer Graphics International conference, CGI 1999*, 1999, pp. 85–92.
- [17] J. Zhang and S. You, "Supporting web-based visual exploration of large-scale raster geospatial data using binned min-max quadtree," in *Proceedings of the 22nd international conference on scientific and statistical database management*, ser. SSDBM'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 379–396.
- [18] R. K. Kothuri, S. Ravada, and D. Abugov, "Quadtree and R-tree indexes in Oracle spatial: a comparison using GIS data," in *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '02. New York, NY, USA: ACM, 2002, pp. 546–557.
- [19] D. Xu and M. N. Do, "On the number of rectangular tilings," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 3225–3230, Oct. 2006.
- [20] R. Kutil and D. Engel, "Methods for the anisotropic wavelet packet transform," *Applied and Computational Harmonic Analysis*, vol. 25, no. 3, pp. 295–314, 2008.
- [21] R. Kutil, "The graph structure of the anisotropic wavelet packet transform," in *Proceedings of the 7th international scientific conference devoted to the 25th anniversary of civil engineering faculty and 50th anniversary of technical university Kosice*, May 2002, pp. 41–47.
- [22] —, "Wavelet domain based techniques for video coding," Ph.D. dissertation, Department of Scientific Computing, University of Salzburg, Austria, Jul. 2002.
- [23] R. Kutil and C. Gfrerer, "A generalization of quad-trees applied to shape coding," in *Proceedings of the 18th International Conference on Systems, Signals and Image Processing (IWSSIP 2011)*, Sarajevo, Bosnia and Herzegovina, Jun. 2011, pp. 265–268.
- [24] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.
- [25] D. Taubman and M. Marcellin, *JPEG2000 — Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002.
- [26] I. Ziyot, I. Abdel-Qader, and H. Al-Otum, "Progressive lossy to lossless compression of roi in mammograms: Effects on microcalcification detection," *Integrated Computer-Aided Engineering*, vol. 15, pp. 241–251, 2008.