Transform Coding of Compound Images Using Matching Pursuit

Harald Nautsch
Department of Electrical Engineering
Linköping University, Sweden
Email: harna@isy.liu.se

Jörn Ostermann
Institut für Informationsverarbeitung
Leibniz Universität Hannover, Germany
Email: ostermann@tnt.uni-hannover.de

***Abstract*—Mixed Raster Content (MRC) coding is an efficient way of coding compound images. The layered model used gives rise to missing data in the foreground and background layers. When using a block-based transform for coding, the usual solution has been to fill in the missing data using some form of interpolation. In this paper we instead present a method using matching pursuit to find the transform coefficents. The presented method gives a gain of up to 1 dB on the tested images, compared to common data filling methods.**

I. Introduction

The ITU-T Mixed Raster Content (MRC) document compression standard [1] specifies a multi-layer representation of compound images, i.e. images consisting of a mix of data of different characteristics, e.g. continuous tone images, binary images and computer graphics. A basic three layer approach could code an image as three layers: A foreground layer and a background layer coded as continuous tone images, and a binary mask layer. On decoding, the binary mask layer is used to determine if a pixel in the decoded image should be picked from the background or the foreground. An MRC coder would try to separate a given image into layers and use different standard coding methods on the different layers. Usually this means that holes on the foreground and background layers will have to be filled in some way. In this paper we will instead use matching pursuit to find missing data.

Recently, work has begun by the ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) to standardize tools for Screen Content Coding (video signals containing compound data, for instance scrolling text overlaid on a regular video stream) in the upcoming High Efficiency Video Coding (HEVC) standard [3].

II. Coding Method

Our coder is based on a simplified version of the intra coding method from the H.264 video coding standard [2], using only the transform block size 8×8. Instead of having full resolution foreground, background and mask layers, we will decide for each block if it is coded using one or three layers.

An 8×8 block of pixels that are about to be coded is first predicted from the pixels surrounding the block (figure 1). There are 9 different ways (modes) of calculating the prediction block. After prediction, the prediction error is transformed using a DCT, quantized and then coded using CAVLC. Each block will either be coded as a single block, corresponding to the normal H.264 intra coding method, or as a mixed block. For a mixed block, we will code a binary mask, a dark block and a bright block. Each of these two blocks will be coded using prediction and transform coding. A decoded mixed block is formed by combining the dark and the bright blocks, using the binary mask to decide from which block the reconstructed pixel is chosen.

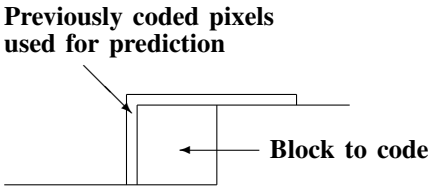


Fig. 1: Block prediction.

A. Early classification

We make a first classification of each block to determine if it is a flat block or not. A flat block is always coded as a single block. For a block that is not flat we will try to code it both as a single block and as a mixed block and then choose coding method using rate-distortion optimization.

The classification method works by optimizing a two-level quantizer for the block. If the difference in distortion between quantizing the block using the two-level quantizer and quantizing the whole block to just one of the levels is small, the block is considered to be flat.

B. Mask

For a block that is not flat, we find a binary mask by using the same two-level quantizer that we used in the early classification. Each pixel of the block will thus be classified as either a dark or a bright pixel. The binary mask is coded using a binary arithmetic coder, with probabilities conditioned on the surrounding three mask pixels.

C. Prediction

For flat blocks the prediction is straight-forward. We will try all 9 different prediction modes and then after transformation,

quantization and variable-length coding choose the prediction mode that gives the lowest coding cost.

For non-flat blocks, we try both the single block coding explained above and mixed coding. For mixed blocks, the dark and the bright pixels will be predicted separately. We will use the surrounding reconstructed values of the corresponding class (dark or bright) if available, i.e. if the surrounding blocks have been coded as mixed blocks. If the surrounding blocks have been coded as single blocks, we will instead use those reconstructed pixel values for prediction. All $9 + 9$ prediction modes are tested for later rate-distortion optimization.

D. Transform

The prediction error blocks are then transformed using a separable DCT. For single blocks this is just a straightforward transformation. For mixed blocks, we have two prediction error blocks that are no longer square blocks of pixels and we can not just use the DCT straight away. One way of solving this is to fill in the missing pixel values of the dark and the bright blocks in some way (see for instance [7] and [5]) and then do a normal DCT. These block filling methods strive to give interpolated blocks that are flat, i.e. have mostly low frequency content.

Instead of first filling in the missing pixel and transforming the blocks, we will find the transform blocks directly given the available pixels. Our goal is to find transform blocks that are sparse, i.e. blocks that have a low number of large transform coefficients. For that reason, we will use a matching pursuit [6] algorithm to find the transform coefficients for each dark and bright prediction error block. Matching pursuit is an iterative process. We start with an all-zero transform block. In each iteration, we will change one transform coefficient so that we maximize the reduction in mean square error between the inverse transformed block and the dark or bright pixel block. The error is only measured for the pixel positions that belong to the dark or bright block. We iterate until the mean square error is below a given threshold T_m. A larger threshold gives a sparser transform block, but will on the other hand introduce more distortion. For our later experiments we have chosen $T_m = 1$.

E. Quantization and coding

Each transform block is quantized uniformly. The quantized coefficients are ordered in zig-zag scan order and then coded using CAVLC just like in H.264. The choice of quantization steps is used to control the rate and distortion we get.

F. Rate-distortion optimization

The cost function for coding a mixed block using two transform blocks is

$$J_2 = 1 + R_b + R_d + R_m + \lambda D_2$$

where R_b and R_d are the number of bits for the bright and dark regions of the block respectively (bits to code the prediction modes and bits from the CAVLC coding of transform coefficients). R_m is the number of bits for coding the mask and D_2 is the distortion between the original pixel block and the reconstructed pixel block. The extra 1 is for a flag bit that tells us if the block is coded as a single block or a mixed block. We will calculate this cost for all possible prediction modes, and choose the modes that give the lowest cost. Since the cost for the dark and the bright areas are independent of each other, we do not have to try all 81 combinations of prediction modes for a mixed block. Instead we can optimize each part separately. In addition, we will also always try to code every block using a single prediction block and a single transform block, even if the mask suggests otherwise. This will give a coding cost of

$$J_1 = 1 + R_1 + \lambda D_1$$

where R_1 and D_1 are the bits and the distortion from single block coding. If $J_2 < J_1$, we will code the block using mixed block coding, otherwise we will use single block coding. For flat blocks we have only tried single block coding and will just choose the prediction mode that gives the lowest cost J_1.

III. Experimental Results

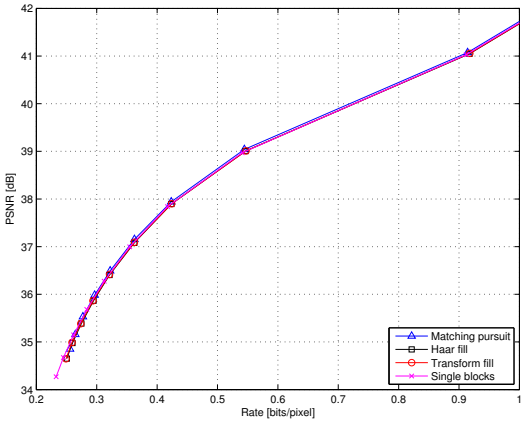
The proposed coder was tested on three grayscale images (see figures 2a, 2c and 2e). Each image is a single frame taken from the sequences used for testing screen content coding in the upcoming HEVC standard [4].

For comparison, two block filling methods were also implemented and tested in our coder simply by replacing the matching pursuit step. The first method is proposed by de Queiroz [7] and works by initializing the missing pixels with the mean value of the available pixels. The block is then iteratively transformed, quantized and inverse transformed to get new pixel values for the missing positions. The algorithm will converge after very few iterations. The second block filling method is proposed by Lakhani and Subedi [5]. It works by interpolating missing pixels in the Haar wavelet domain, such that the sum of the squares of the AC wavelet coefficients is minimized. To get an idea of how much can be gained by mixed block coding, the three different coders are also compared with a coder that only uses single block coding, i.e. no blocks are coded as mixed blocks.

The results for each test image can be seen in figures 2b, 2d and 2f. For image 1, approximately 5% of the blocks are coded as mixed blocks. For images 2 and 3, approximately 15% and 30% of the blocks are coded as mixed blocks, respectively. For all three images, our matching pursuit coding method outperforms the two block filling algorithms. For image 1, the gain from using matching pursuit is around 0.05 dB, but for this particular image the mixed block coding gives very little gain compared to single block coding. For image 2 the gain from using matching pursuit ranges from 0.1 dB for low rates to 0.3 dB for high rates. For image 3 the gain is around 1 dB. Compared to single block coding, mixed block coding gives a substantial gain for images 2 and 3.

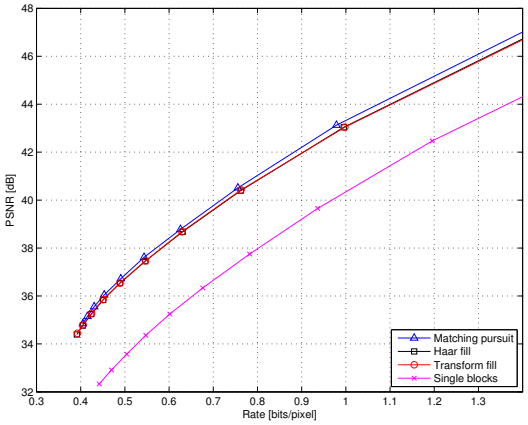
In figure 3 is shown an example of what the different layers look like for a small part of one of the test images, using matching pursuit. As can be seen in images 3d and

(a)

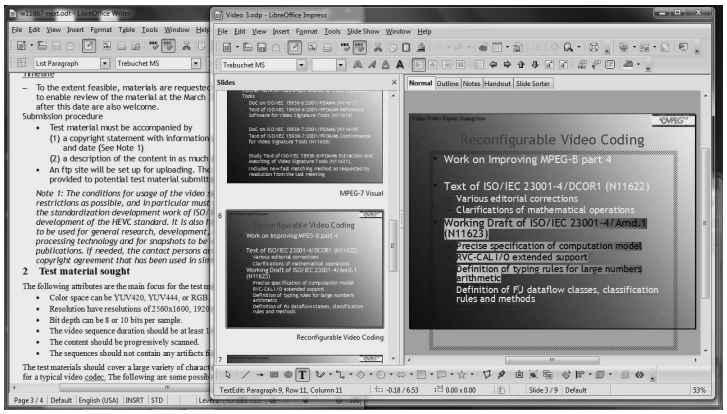


(b)

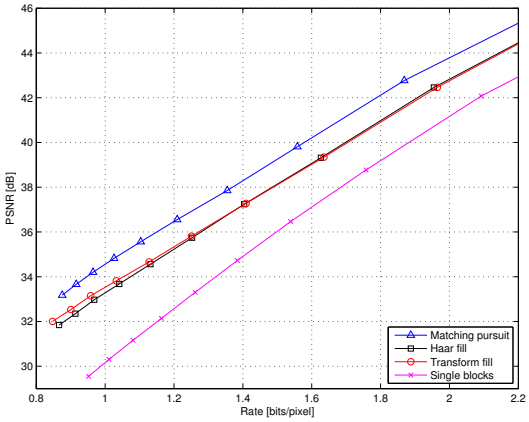
(c)



(d)



(e)



(f)

Fig. 2: Three test images and corresponding rate-PSNR plots.

3e, the matching pursuit process might give blocks with high frequency content.

IV. Conclusion

We presented a method for layered transform coding of images, using matching pursuit to find transformed blocks instead of previous methods using data filling before the transform. The proposed method was compared to two block filling methods and was found to give a gain of between 0.1 dB and 1 dB, depending on the images.

References

[1] "Mixed Raster Content (MRC)", ITU-T recommendation T.44, Study Group 8, 1997.

[2] "Final Draft International Standard of Joint Video Specification", ITU-T recommendation H.264, ISO/IEC 14496-10 AVC, 2003.

[3] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, T. Wiegand (editors) "D4: Working Draft 4 of High Efficiency Video Coding", 6th JCT-VC Meeting, Torino, Italy, Doc. No. JCTVC-F803, July 2011.

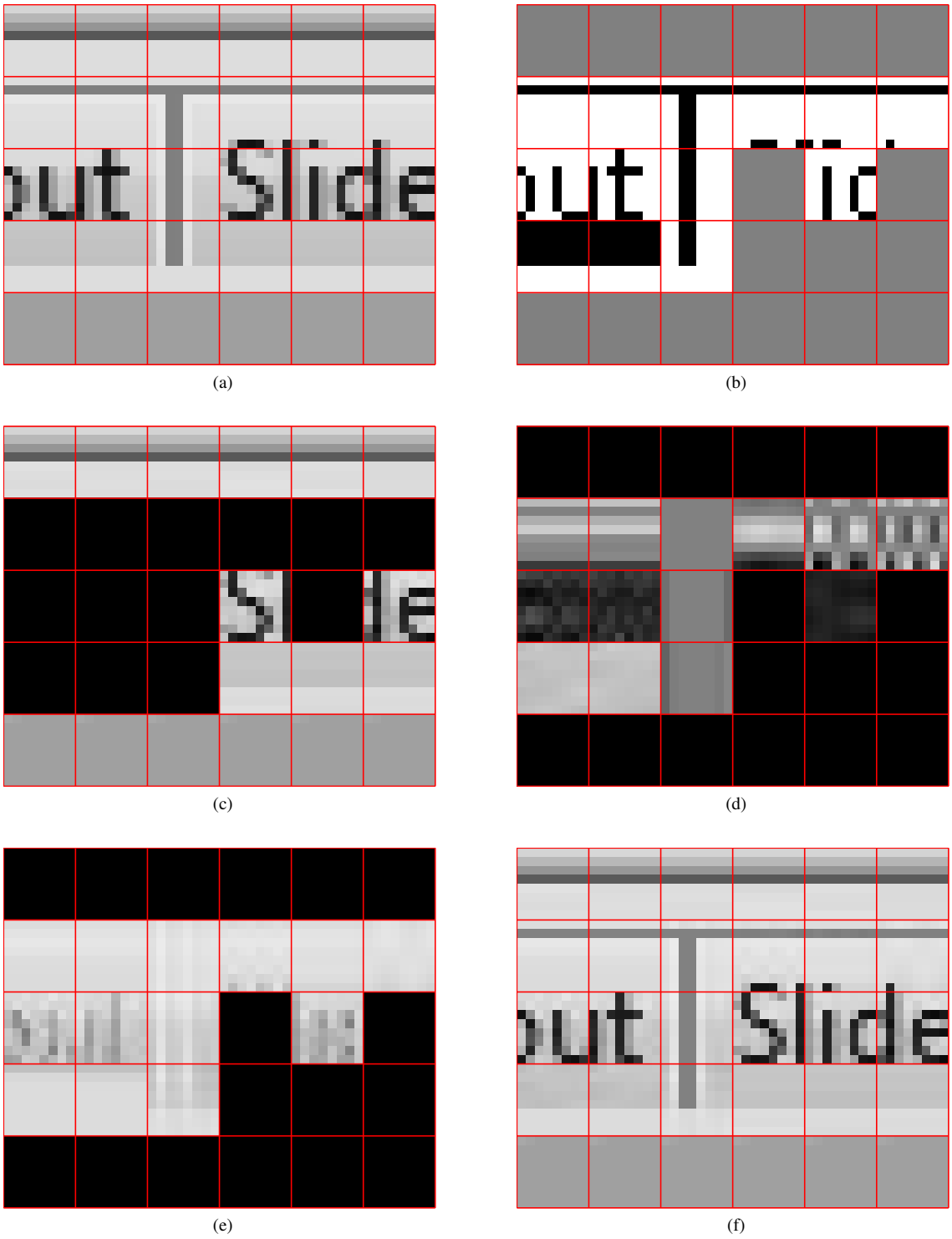


Fig. 3: A small part of test image 3. Block borders have been marked in red. (a) Original image. (b) Classification and binary mask. Single blocks are gray, mixed blocks have their dark and bright regions in black and white, respectively. (c) Decoded single blocks. (d) Decoded dark blocks. (e) Decoded bright blocks. (f) Resulting decoded image.

[4] O. C. Au, J. Xu, H. Yu, "BoG report on Screen Content Coding (SCC)", 6th JCT-VC Meeting, Torino, Italy, Doc. No. JCTVC-F771, July 2011.

[5] G. Lakhani and R. Subedi, "Optimal Filling of FG/BG Layers of Compound Document Images", *Proc. IEEE Intl. Conf. on Image Processing*, pp. 2273-2276, 2006.

[6] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries", *IEEE Trans. Signal Processing*, vol. 41, pp. 3397-3415, 1993.

[7] R.L. de Queiroz, "On Data Filling Algorithms for MRC Layers", *Proc. IEEE Intl. Conf. on Image Processing*, pp. 586-589, 2000.