Hamburger Beiträge
zur Angewandten Mathematik

Contextual Image Compression from Adaptive Sparse Data Representations

Laurent Demaret, Armin Iske, Wahid Khachabi

Nr. 2009-05
May 2009

Contextual Image Compression from Adaptive Sparse Data Representations

Laurent Demaret
HelmholtzZentrum München
Institute of Biomathematics and Biometry
Ingolstädter Landstraße 1
D-85764 Neuherberg, GERMANY
laurent.demaret@helmholtz-muenchen.de

Armin Iske
University of Hamburg
Department of Mathematics
Bundesstraße 55
D-20146 Hamburg, GERMANY
iske@math.uni-hamburg.de

Wahid Khachabi
University of Hamburg
Department of Mathematics
Bundesstraße 55
D-20146 Hamburg, GERMANY
khachabi@gmail.com

***Abstract*—Natural images contain crucial information in sharp geometrical boundaries between objects. Therefore, their description by smooth isotropic function spaces (e.g. Sobolev or Besov spaces) is not sufficiently accurate. Moreover, methods known to be optimal for such isotropic spaces (tensor product wavelet decompositions) do not provide optimal nonlinear approximations for piecewise smooth bivariate functions. Among the geometry-based alternatives that were proposed during the last few years, *adaptive thinning* methods work with continuous piecewise affine functions on anisotropic triangulations to construct sparse representations for piecewise smooth bivariate functions. In this article, a customized compression method for coding the sparse data information, as output by adaptive thinning, is proposed. The compression method is based on contextual encoding of both the sparse data positions and their attached luminance values. To this end, the structural properties of the sparse data representation are essentially exploited. The resulting contextual image compression method of this article outperforms our previous methods (all relying on adaptive thinning) quite significantly. Moreover, our proposed compression method also outperforms JPEG2000 for selected natural images, at both low and middle bitrates, as this is supported by numerical examples in this article.**

I. Introduction

Many applications in image analysis require a preprocessing decomposition of the underlying function over a specific library of building blocks, which are usually depending on the nature of the image. For the sake of efficiency, the utilized representation is required to be sparse. Moreover, the image reconstruction, to be obtained from a small number of atoms, should match further specific requirements of the diverse target applications, such as object segmentation, denoising, and geometry-preserving compression. In the latter, a challenging task is the design of effective edge-preserving methods for image compression. It has become a widespread paradigm to model the image geometry mathematically by bivariate functions which are smooth over a finite number of (two-dimensional) domain regions separated by (one-dimensional) locally regular curves, see [6].

Recall that JPEG2000 (based on [10]) works – as well as other efficient image compression methods – with a wavelet decomposition of the image, followed by a suitable encoding method. In the utilized encoding methods, the clustering of significant image coefficients along sharp edges (or other characteristic features) is exploited, so that in large smooth areas of the image only very few coefficients are needed in their representation. As shown in [7], however, the representation of characteristic functions over smooth domains by wavelet coefficients has severe restrictions. In fact, the corresponding nonlinear approximations, as they are obtained by a simple thresholding of the wavelet coefficients, are only suboptimal. Just very recently, different improvements using adaptive triangulations have been developed to tackle this problem, see [1], [8], [9], [12].

This article proposes an alternative concept for *contextual compression* of geometry-dominated images. The resulting compression method relies on adaptive image approximation using *adaptive thinning*, which yields a sparse representation of the image through a very small scattered set of *significant pixels*. The adaptive thinning algorithm, dating back to our papers [2], [3], is a recursive point removal scheme for scattered data, which has recently resulted in a complete image compression method [3], [4]. The selection of the significant pixels by adaptive thinning works with linear splines over adaptive Delaunay triangulations. The resulting compression method was shown to outperform JPEG2000 for a test set of natural and geometrical images [3], [4].

But the compression methods in [3], [4] do not rely on contextual-based coding. Instead of this, they work with a rather pragmatic and simple hierarchical coding scheme for scattered data, dating back to our 2003 paper [5]. Now the present paper improves the compression methods of our previous work quite significantly by using a completely different concept for the coding. In fact, this paper is the first one, where we combine adaptive thinning (for image approximation) with contextual coding. Unlike in [3], [4], the focus of this paper is on coding rather than on approximation.

To be somewhat more precise, the contextual-based coding scheme of this paper essentially exploits (unlike in [3], [4]) the structure of our adaptive image approximation, where spatial properties of the utilized anisotropic triangulations play a key role: due to their construction (by adaptive thinning), long, thin and steep triangular elements are aligned with sharp boundaries between neighbouring objects, whereas large, wide and flat triangles are lying in smooth areas. In the design of suitable local coding contexts, we make essential advantage of this important point.

In comparison with contextual coding in wavelet-based compression methods [10], the coding of the wavelet decomposition is done by using judiciously chosen local contexts: spatial structures in the wavelet domain are exploited to reduce the size of the compressed image. This explains that in spite of non-optimal approximation rates for wavelets, JPEG2000 outputs good image reconstructions at high compression rates.

In the numerical comparisons of Section III, we show that the performance of the proposed contextual-based compression method reduces the coding costs of our previous compression methods [3], [4] quite significantly. Moreover, it is shown that our contextual-based compression method outperforms JPEG2000 now also for higher (more realistic) compression rates, unlike the compression methods in [3], [4].

II. Contextual Compression of Sparse Image Data

A. Image Approximation and Sparse Data Representation

We consider a digital gray-scale image

$$I : X = [1, N] \times [1, M] \to [0, 1, \ldots, 2^r - 1]$$

as a function, where $X \subset \mathbb{Z}^2$ is the discrete pixel domain of the image function I, and r is the number of bits used for the representation of the luminance values. The adaptive thinning algorithm [3] constructs a small subset $Y \subset X$ of *significant pixels* in only $\mathcal{O}(|X| \log(|X|))$ steps, where $|X|$ is the number of pixels in X. The (unique) Delaunay triangulation $\mathcal{D}(Y)$ of Y yields a linear approximation space S_Y, containing all continuous linear spline functions over $\mathcal{D}(Y)$.

Now the resulting sparse image representation is given by the (unique) best approximating spline function $L_Y^* \in S_Y$ which minimizes the mean square error among all linear splines in S_Y. The construction of Y, by adaptive thinning, and so the selection of the spline space S_Y, aims to reduce the resulting minimal mean square error. Since the selection of an *optimal* subset $Y^* \subset X$ (among all subsets $Y \subset X$ of equal size) is known to be an NP-hard optimization problem, the adaptive thinning algorithm could be viewed as a very efficient method for finding a good image approximation in a (huge) dictionary of linear splines over triangulations.

The sparse representation of image I can be written as

$$L_Y^*(p) = \sum_{q \in Y} L_Y^*(q)\varphi_q(p) \qquad \text{for } p \in X,$$

where the *Courant element* $\varphi_q \in S_Y$ is, for $q \in Y$, the unique Lagrangian basis function satisfying

$$\varphi_q(p) = \begin{cases} 1 & \text{for } q = p; \\ 0 & \text{for } q \neq p; \end{cases} \qquad \text{for } q \in Y.$$

Since the Delaunay triangulation $\mathcal{D}(Y)$ of the significant pixels $Y \subset X$, as output by adaptive thinning, will be unique (see [3], [4] for subtle details), there is no need to code any connectivity information. Therefore, we are concerned with coding sparse data from the small information set

$$\{(p, \alpha_p) \,:\, p \in Y \;\text{ and } \alpha_p \in \{0, 1, \ldots, S\}\},$$

where $S < 2^r$, due to uniform quantisation.

Now our proposed compression algorithm performs the following two steps, one after the other.

(i) code the subset Y (see Subsection II-C);
(ii) code the symbols α_p, for $p \in Y$ (see Subsection II-D).

At the decoder, the information Y is decoded first, before the corresponding Delaunay triangulation $\mathcal{D}(Y)$ is reconstructed from Y. Therefore, the required connectivity information is readily available at step (ii), and so it can be used for the purpose of contextual coding.

B. Deterministic Contextual Encoding

Let us first recall some general principles of contextual encoding. Contextual encoding of an information unit makes use of *causal information* being attached to this unit. The causal information is usually given by specific context information that is available at the corresponding decoding step. A *context* is given by causal information that is used by the encoder and the decoder to agree on the mode according to which a current symbol is coded.

For the sake of further illustration, let us make one simple example. Suppose the symbols to be coded are binary (0 or 1) and there are only two possible coding modes (say A and B) determined by the context. A contextual compression scheme can be regarded as a map which associates each context of a symbol with a mode. With that map, the occurences of 0 and 1, conditionally to the modes A and B, are first counted. Let us denote the occurences by $n_{0,A}$, $n_{1,A}$, $n_{0,B}$ and $n_{1,B}$ (e.g. $n_{0,A}$ may be the number of 0 to be coded according to the coding mode A). Each symbol is then coded according to its corresponding conditional frequency in the set of symbols which have not been coded, yet.

We remark that non-adaptive arithmetic encoding usually produces codes whose length is given by the *conditional entropy*, defined as

$$\log\left(\frac{n_{0,A}}{n_{0,A} + n_{1,A}}\right) + \log\left(\frac{n_{0,B}}{n_{0,B} + n_{1,B}}\right) < \log\left(\frac{n_0}{n_0 + n_1}\right),$$

where the upper bound on the right hand side is the global (non-contextual) entropy of the data. Contextual encoding decreases the corresponding coding length quite significantly, especially when the differences between the conditional frequencies $n_{0,A}/(n_{0,A}+n_{1,A})$ and $n_{0,B}/(n_{0,B}+n_{1,B})$ are large.

Now we consider the generic case where the symbols to be coded belong to a finite set $\mathcal{S}$ and the contextual modes belong to a finite set $\mathcal{M}$. The combinatorial contextual coding method is formulated in Algorithm 1.

In contrast to standard contextual encoding methods, where adaptive (i.e. learning probabilistic) approaches are used [10], we prefer to work with a combinatorial (i.e. deterministic) approach, where the frequencies are initially transmitted (in step (1)) and updated after each symbol coding (in step (3c)).

In the following two subsections we design contextual modes which lead to small conditional entropies for the two relevant classes of information sets, that is (i) the positions of the significant pixels and (ii) the luminance values at these pixels. Since image structures are local in space, *causal context* is, for any element to be coded, given by causal information restricted to its local neighbourhood.

Algorithm 1: **(Combinatorial Contextual Coding).**

(1) Compute frequency table $n_{s,M}$ for all $s \in \mathcal{S}$ and $M \in \mathcal{M}$; then transmit n_{ij};

(2) Add the table to the bitstream;

(3) For $i = 1, \dots, N$;

  (3a) Determine the contextual state M of i;

  (3b) Code symbol s_i according to a deterministic arithmetic coding scheme, where each symbol $s \in \mathcal{S}$ is assigned a probability $p_s = n_{s,M}/(\sum_{t \in S} n_{t,M})$;

  (3c) Let $n_{s,M} = n_{s,M} - 1$;

C. Contextual Coding of the Pixel Positions

This section deals with the coding of the pixel positions, or, equivalently, with the coding of the subset $Y \subset X$ of significant pixels, as output by adaptive thinning (step (i)). We recall that adaptive thinning outputs clusters of significant pixels which are aligned with sharp edges (i.e. regular curves). But significant pixels are also clustered around fine details of the image, see the numerical examples [3], [4]. We make use of these observations to design appropriate contexts.

For the sake of simplicity, we restrict ourselves to the case where the information (i.e. the significant pixels) are coded line by line, from top-left to bottom-right. Accordingly, we define a *contextual box of order* r, $B(p)$, of a pixel $p = (p_x, p_y) \in X$ as the set of causal pixels q satisfying $\max(|p_x - q_x|, |p_y - q_y|) \leq b$, for $b \in \mathbb{N}$. For any pixel $p \in X$, we denote by n_p the number of significant pixels in $B(p)$, i.e. $n_p = |Y \cap B(p)|$. The numbers n_p are then used to determine the contextual mode of the point p. We remark that the size of the boxes and the number of modes become parameters of the coding method which have to be transmitted in the global header. An illustrating example is displayed in Figure 1.

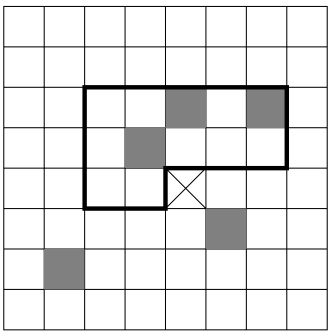

Fig. 1. Contextual Box. Each square of the grid corresponds to a pixel p; significant pixels are displayed in gray. The crossed square is the pixel to be coded. The contextual box for $b = 2$ is the area surrounded by the fat line; it contains $n_p = 3$ significant pixels.

D. Contextual Coding of the Luminance Values

We remark that the Delaunay triangulation $\mathcal{D}(Y)$ of the significant pixels $Y \subset X$ – output by adaptive thinning – can be used as causal context in the coding/decoding of the (gray-scale) luminance values. We can explain this as follows.

For piecewise smooth functions, large, wide and flat triangles are lying in smooth areas, whereas long, thin and steep triangles are lying near the image contours. This is due to the construction of the triangulation $\mathcal{D}(Y)$ by adaptive thinning, which aims to capture the geometry of the image. Therefore, for pixel pairs lying opposite across an edge of $\mathcal{D}(Y)$ the difference between their luminances is assumed to be a decreasing function of the edge length. This assumption is corroborated by numerical observations, see the example `Drunken Sunset` in Figure 2.

This reasonable assumption motivates our iterative contextual coding strategy: starting at the longest edge in $\mathcal{D}(Y)$, we transmit the differences of the luminances along the current *longest neighbouring edge*, respectively, for which the information at the other vertex has already been coded. The range taken by the lengths of the longest causal edge is partitioned into intervals each of which corresponding to one contextual mode.

III. NUMERICAL RESULTS

In this section, we compare the performance of the proposed contextual image compression method, **CAT**, with that of our previous compression method, **AT** in [3], and with that of JPEG2000 [10]. For JPEG2000 we used the Kakadu implementation available at [11]. The results presented for **CAT** and **AT** are based on a full compression/decompression scheme rather than on an evaluation of theoretical coding costs.

The utilized test cases are comprising three popular images: `Cameraman`, `Lena` and `Fruits`. The `Cameraman` test image is of size 256×256 pixels, whereas the other two test images are containing 512×512 pixels.

Figure 3 shows rate-distortion curves for the three test cases, where for each of the three different compression methods the resulting *peak signal to noise ratio* (PSNR, measured in `dB`) is plotted as a function of the compressed data size (measured in *bits per pixel*, `bpp`).

Note that in all test cases, the compression method **CAT** of this article outperforms our previous compression method **AT** quite significantly. Moreover, the performance of **CAT** is also superior to JPEG2000 for a larger range of small and mid bitrates, unlike **AT**. For the geometry-dominated test image `Fruits`, for instance, the compression method **CAT** outperforms JPEG2000 for all bitrates smaller than 0.6 `bpp`. For the test image `Lena`, **CAT** outperforms JPEG2000 for all bitrates smaller than 0.27 `bpp`. But for bitrates larger than 0.27 `bpp`, JPEG2000 achieves to outperform **CAT**. This is due to textures in the test image `Lena` whose accurate resolution, when working with **CAT**, requires more energy (in terms of significant pixels), whereas the wavelet-based JPEG2000 renders higher frequencies (like textures) very well for high bitrates.

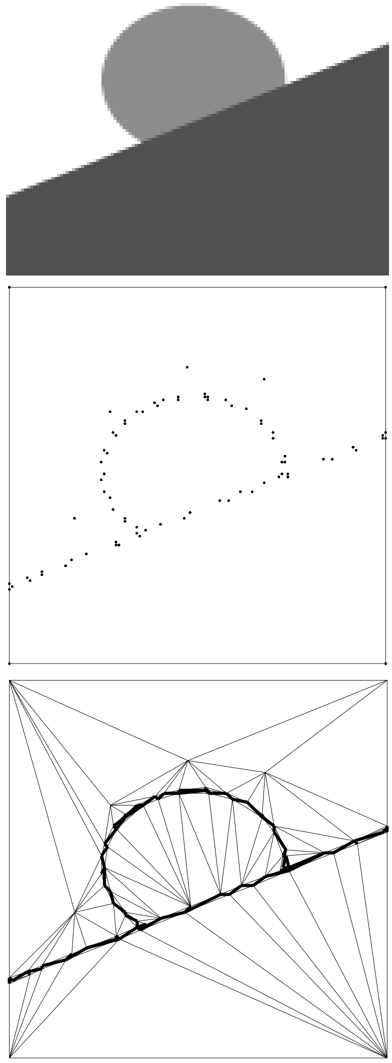

Fig. 2. `Drunken Sunset`. Original image (top); 80 significant points Y (middle); Delaunay triangulation $\mathcal{D}(Y)$ (bottom). Edges with high luminance variation are fat.

In conclusion, we can recommend the proposed compression method **CAT** (also our previous **AT**) for images with dominant geometrical features and with widely texture-free content. For texture-dominated images, however, our compression method **CAT** can only be competitive at small bitrates.

Figures 4–6 show visual comparisons for three selected test cases. Note that for all test images, our compression method **CAT** avoids, unlike the wavelet-based JPEG2000, oscillating artefacts around edges, especially for the test image `Cameraman`. As supported by the presented test cases, our compression method **CAT** is edge-preserving and denoising.

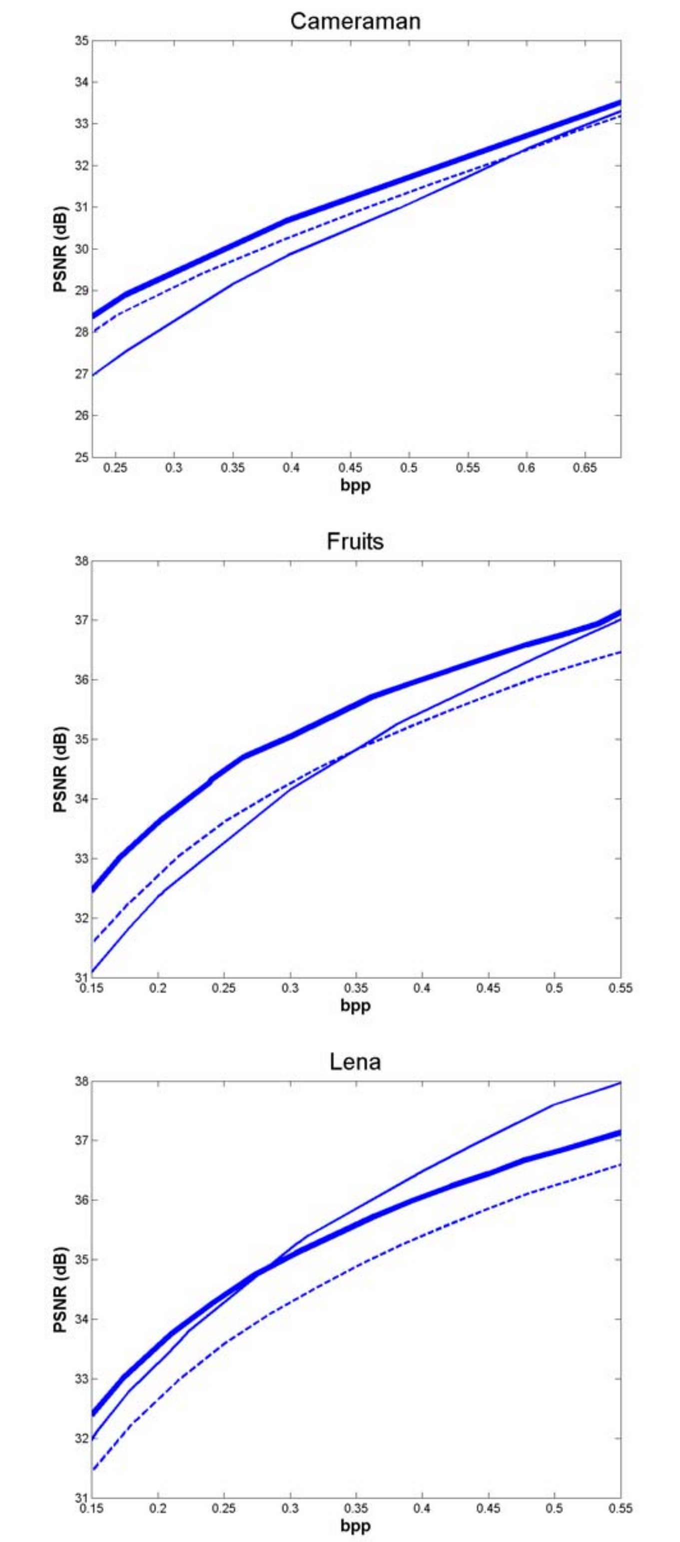

Fig. 3. Comparison of rate-distortion curves (bits per pixel vs PSNR) for methods JPEG2000 (thin curve); **AT** of [3] (dashed curve); **CAT** (of this article, fat curve). Three images are tested: `Cameraman`, `Fruits` and `Lena`.

JPEG2000.
PSNR: 29.84 dB at 3247 Bytes

Contextual Compression.
PSNR: 30.66 dB at 3233 Bytes

Fig. 4. Test image Cameraman. Comparison between JPEG2000 and the contextual compression method **CAT**.

References

[1] A. Cohen, N. Dyn, F. Hecht, J.-M. Mirebeau: *Adaptive multiresolution analysis based on anisotropic triangulations*. Preprint.

[2] L. Demaret, N. Dyn, M. S. Floater, and A. Iske: *Adaptive thinning for terrain modelling and image compression*. In: Advances in Multiresolution for Geometric Modelling, N. A. Dodgson, M. S. Floater, and M. A. Sabin (eds.), Springer-Verlag, Heidelberg, 2004, 321–340.

[3] L. Demaret, N. Dyn, A. Iske: *Image compression by linear splines over adaptive triangulations*. Signal Processing **86**(7), July 2006, 1604–1616.

[4] L. Demaret, A. Iske: *Adaptive image approximation by linear splines over locally optimal Delaunay triangulations*. IEEE Signal Processing Letters **13**(5), May 2006, 281–284.

[5] L. Demaret and A. Iske: *Scattered data coding in digital image compression*. In: Curve and Surface Fitting: Saint-Malo 2002, A. Cohen, J.-L. Merrien, and L. L. Schumaker (eds.), Nashboro Press, Brentwood, 2003, 107–117.

[6] J. Garnett, T. Le, Y. Meyer, L. Vese: *Image decompositions using bounded variation and generalized homogeneous Besov spaces*. Appl. Comput. Harmon. Anal. **23**, 2007, 25–56.

[7] S. Jaffard: *Beyond Besov spaces, part 2: oscillation spaces*. Constr. Approx. **21**, 2005, 29–61.

[8] B. Lehner, G. Umlauf, B. Hamann: *Video compression using data-dependent triangulations*. In: Computer Graphics and Visualization '08, Y. Xiao, E. ten Thij (eds.), 2008, 244–248.

[9] M. Sarkis and K. Diepold: *Geometric compression for interactive transmissions*. IEEE Trans. Im. Proc., in press.

[10] D. Taubman: *High performance scalable image compression with EBCOT*. IEEE Trans. on Image Processing, July 2000, 1158–1170.

[11] D. Taubman: http://www.kakadusoftware.com/

[12] W. Van Aerschot, M. Jansen, A. Bultheel: *Normal Mesh based Geometrical Image Compression*. IMAVIS, in press.

JPEG2000. PSNR: 34.15 dB at 9829 Bytes

Contextual Compression. PSNR: 35.04 dB at 9825 Bytes

Fig. 5. Test image Fruits. Comparison between JPEG2000 and the contextual compression method **CAT**.

JPEG2000. PSNR: 33.56 dB at 6986 Bytes

Contextual Compression. PSNR: 33.75 dB at 6873 Bytes

Fig. 6. Test image Lena. Comparison between JPEG2000 and the contextual compression method **CAT**.