

# Advances in Digital Image Compression by Adaptive Thinning

Laurent Demaret (demaret@ma.tum.de) and Armin Iske (iske@ma.tum.de)

Zentrum Mathematik, Technische Universität München, D-85747 Garching, Germany

## Abstract

This paper proposes a novel concept for digital image compression. The resulting compression scheme relies on adaptive thinning algorithms, which are recent multiresolution methods from scattered data approximation. Adaptive thinning algorithms are recursive point removal schemes, which are combined with piecewise linear interpolation over decremental Delaunay triangulations. This paper shows the utility of adaptive thinning algorithms in digital image compression. To this end, specific adaptive pixel removal criteria are designed for multiresolution modelling of digital images. This is combined with a previous customized coding scheme for scattered data. The good performance of our compression scheme is finally shown in comparison with the well-established wavelet-based compression method SPIHT.

## 1. INTRODUCTION

Over the past few years, digital image compression has become a major challenge in information technology. For the efficient transmission of digital image data it is crucial to reduce the amount of information by using a sparse representation of the image. To this end, effective coding schemes are used in order to maintain a good approximation quality at low bitrates. For a comprehensive introduction to various relevant aspects of digital image compression, we recommend the textbook [8].

The performance of any image compression scheme depends on its ability to capture characteristic features of the image, such as sharp edges and fine textures, while reducing the number of parameters used for its modelling. Most of the well-established compression schemes are using the bivariate Discrete Wavelet Transform (DWT), see the survey [2] on wavelet-based image coding. At high compression rates, wavelet-based methods provide much better image quality in comparison with the JPEG standard, which relies on the Discrete Cosine Transform (DCT). The good results obtained from DWT are due to sophisticated techniques which essentially take advantage of the statistical structure of the image data.

For instance, the well-established method SPIHT [6] (Set Partitioning Into Hierarchical Trees) uses judicious clusters of non-significant coefficients (collected in *zerotrees*) in order to efficiently encode the most significant coefficients. The new compression standard JPEG2000 (see [8]), based on EBCOT [7], uses contextual encoding, which models the Markovian structures in the pyramidal wavelet decomposition of the image. At very low bit rates, however, the oscillatory behaviour of wavelet bases typically

leads to undesirable artefacts along sharp edges. Wavelets require too many non-vanishing coefficients to represent sharp edges accurately, because they introduce artificial oscillations in their modelling.

This paper proposes an alternative concept for digital image compression, which is particularly well-suited for the modelling of sharp edges and related features in digital images. To this end, piecewise linear functions over adaptive triangular meshes are used. This leads to a reduction of low-pass filtering effects, which are often due to overquantization of the high-pass coefficients. The modelling relies on recursive point removal schemes, termed *adaptive thinning algorithms*, and a customized coding scheme for scattered pixels.

The outline of the paper is as follows. The application of adaptive thinning algorithms to digital image modelling is explained in Section 2. Then, in Section 3, the above-mentioned coding scheme is briefly discussed. Finally, selected numerical examples are shown in Section 4, where the performance of our compression scheme is compared with the wavelet-based compression method SPIHT.

## 2. ADAPTIVE THINNING IN IMAGE MODELLING

In order to keep this paper widely self-contained, this section first introduces basic concepts and ingredients of adaptive thinning algorithms, before their application to digital image modelling is discussed.

In many classical image compression methods, such as for the aforementioned DCT and DWT, the modelling is carried out by decomposing the image over a non-adaptive orthogonal basis of functions. The corresponding coefficients of the basis functions are then quantized, according to a specific quantization step, which usually depends on a target compression rate. The performance of the resulting compression scheme depends on the approximation quality which results from the non-vanishing coefficients. We remark that such approaches do not allow any adaptivity in the choice of the representation functions. The high quantization steps required for very high compression rates lead to undesirable low-pass filter artefacts, also called ringing artefacts. Visually, they correspond to oscillations around sharp edges. Examples of such artefacts are shown in Figure 1, where the well-known test image Goldhill is used.

An alternative modelling concept represents the image by piecewise linear functions over triangular meshes. The resulting approaches based on hierarchies of *regular* (i.e. non-adaptive) triangular meshes, however, lead to low-pass



Figure 1: Goldhill. Coding by using SPIHT at a very high compression rate of 0.125 bpp (bits per pixel) leads to ringing artefacts.

artefacts, such as in the case of wavelets. In contrast, *irregular* adaptive triangulations offer much more flexibility, and they support the appropriate concept of adaptivity for representing natural features of the image. In view of the required compression, however, this enhanced flexibility may lead to high coding costs required for coding the node coordinates, and the topology coding (connectivity between nodes) of the corresponding mesh.

In order to entirely avoid the required costs for the connectivity coding, adaptive thinning algorithms work with Delaunay triangulations. Recall that a Delaunay triangulation of a discrete planar point set  $X \subset \mathbb{R}^2$  is a triangulation, such that the circumcircle of each of its triangles does not contain any point from  $X$  in its interior. An example of such a triangulation is shown in Figure 2. For further details concerning triangulation methods, we refer to the textbook [5].

Now we associate, with any finite set  $X$  of points its *unique* Delaunay triangulation  $D_X$  (in the case of co-circular points in  $X$  there may be ambiguities which we exclude in the following discussion for the sake of simplicity). Thus, at the decoder, the set of points  $X$  can directly be used in order to uniquely reconstruct the triangulation  $D_X$ .

The adaptive thinning algorithm, first developed in [4], is concerned with the approximation of a bivariate function  $f$  from a finite set of scattered data points  $X \subset \mathbb{R}^2$  and sample values  $\{f(x)\}_{x \in X}$ . To this end, a data hierarchy

$$X = X_N \supset X_{N-1} \supset \dots \supset X_n \quad (1)$$

of nested subsets of  $X = \{x_1, \dots, x_N\} \subset \mathbb{R}^2$  is constructed. This is done by recursively removing points from  $X$ . At any removal step, one point is removed from the current subset  $X_p \subset X$  in (1), so that  $X_p$  is of size  $|X_p| = p$ ,  $n \leq p \leq N$ .

The multiresolution method, associated with adaptive thinning, works with decremental Delaunay triangulations over the subsets in (1). To this end, for any subset  $Y \subset X$ ,

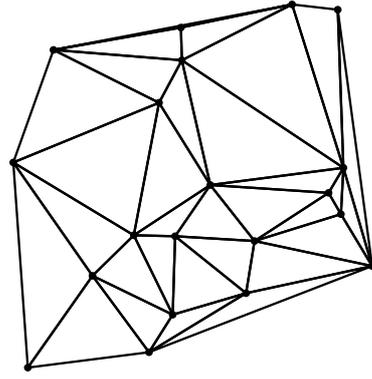


Figure 2: Delaunay Triangulation of a point set.

the target function  $f$  is approximated by the unique continuous function  $L(f, Y)$ , whose restriction on any triangle in the Delaunay triangulation  $D_Y$  is a linear function and which satisfies the interpolation conditions

$$L(f, Y)(y) = f(y), \quad \text{for all } y \in Y.$$

In the subsequent of this text, we say that  $L(f, Y)$  is the piecewise linear interpolant of  $f$  over  $D_Y$ .

Now the aim of adaptive thinning is to remove at each step a point  $x_p = X_p \setminus X_{p-1}$  such that  $L(f, X_{p-1})$  is *close* to the original function  $f$ . In order to design a suitable point removal criterion, we employ a specific norm  $\|\cdot\|$  used to evaluate the approximation error

$$\eta(Y) \equiv \eta(Y, f) = \|L(f, Y) - f\|.$$

The main application addressed in [4] is digital terrain modelling. The criterion used to remove a point from a set is based in [4] on the  $L^\infty$ -norm, measuring the maximal deviation between the original function and the reconstructed function. In this case, we have

$$\eta_\infty(Y) = \|L(f, Y) - f\|_\infty = \max_{x \in X} |L(f, Y)(x) - f(x)|.$$

Our purpose in this article is to apply adaptive thinning to digital images. In this case, the initial set of points  $X$  is given by the set of pixels in the original image. Moreover, the function values of  $f$  are the luminance values at the pixels. In view of digital image compression by adaptive thinning, a suitable removal criterion should be related to the *mean square error* (MSE). Indeed, as supported by numerical examples, this helps to improve the quality of the reduced images, according to the human visual perception.

In this case, we prefer to work with the discrete  $L^2$ -error  $\eta(Y)$ , with respect to the set  $X$  given by

$$\eta^2(Y) = \|L(f, Y) - f\|_2^2 = \sum_{x \in X} |L(f, Y)(x) - f(x)|^2,$$

rather than with the above error measure  $\eta_\infty$ . In order to design a suitable removal criterion for adaptive thinning, we let the *significance* of any point  $y \in Y$  be given by

$$\sigma(y) = \eta^2(Y \setminus y) - \eta^2(Y).$$

Moreover, a point  $y^* \in Y$  is said to be *removable* from  $Y$ , if and only if  $y^*$  is least significant among all points in  $Y$  by satisfying

$$\sigma(y^*) = \min_{y \in Y} \sigma(y).$$

In order to further reduce the resulting computational costs, we restrict our computations for computing the significance of any point  $y \in Y$  to its local cell  $C(y)$  in  $D_Y$ . Recall that the cell  $C(y)$  of the vertex  $y$  is given by the triangles in  $D_Y$  which contain  $y$  as a vertex. This leads us to the simpler significance

$$\sigma(y) = \eta_{C(y)}^2(Y \setminus y) - \eta_{C(y)}^2(Y),$$

where  $\eta_{C(y)}(Y)$  denotes the  $L^2$ -error over the cell  $C(y)$ , i.e.,

$$\eta_{C(y)}^2(Y) = \sum_{x \in C(y) \cap X} |L(f, Y)(x) - f(x)|^2.$$

Having constructed a subset  $X_n$  of  $n$  most significant pixels,  $n \ll N$ , by adaptive thinning, we are in a position to compute a reconstruction of the original image, such as for Lena in Figure 3 (a), at the decoder as follows. The  $n$  most significant pixels, shown in Figure 3 (b) (where  $n = 2205$ ) are used in order to create the corresponding Delaunay triangulation  $D_{X_n}$ , shown in Figure 3 (c). This in turn yields, by evaluating the piecewise linear interpolant  $L(f, X_n)$ , the reconstructed image (Figure 3 (d)).

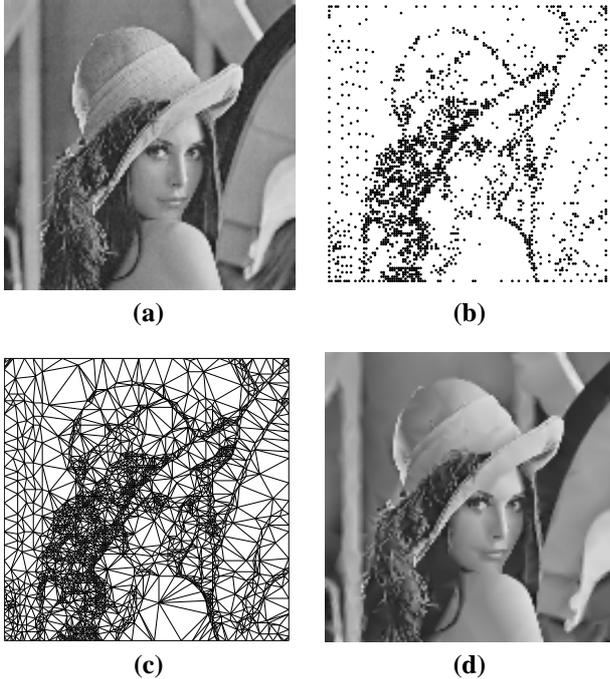


Figure 3: Lena . (a) Original image of size 128-by-128; (b) subset  $X_n$  of  $n = 2205$  most significant pixels; (c) the Delaunay triangulation  $D_{X_n}$ ; (d) reconstructed image.

We remark that the reconstruction quality of the so obtained image, such as in Figure 3 (d), can be further improved. This is done as follows. We consider using *least squares approximation* (LSA) with respect to the approximation space  $V_{X_n} = \text{span}(\varphi_y)_{y \in X_n}$ , where the cardinal

basis function  $\varphi_y$ ,  $y \in X_n$ , is the unique piecewise linear function over  $D_{X_n}$  satisfying

$$\varphi_y(x) = \begin{cases} 1 & \text{for } x = y \in X_n, \\ 0 & \text{for } x \in X_n \setminus y. \end{cases}$$

Due to the properties of the least squares approximation scheme, this yields *optimal* luminance values  $(f^*(y))_{y \in X_n}$  at the pixels in  $X_n$ , so that the corresponding best approximation

$$L^*(f, X_n) = \sum_{y \in X_n} f^*(y) \varphi_y,$$

attains the least squares error by satisfying

$$\|L^*(f, X_n) - f\|_2^2 = \min_{g \in V_{X_n}} \|g - f\|_2^2,$$

where  $\|\cdot\|_2$  is the discrete  $L^2$ -norm with respect to  $X_n$ . For a comprehensive treatment of least squares approximation methods, we recommend the textbook [1].

The use of the criterion based on the  $L^2$ -norm, in combination with least squares approximation, yields a significant improvement of the image quality, as shown in Figure 4 for the test image Barbara of size 256-by-256.

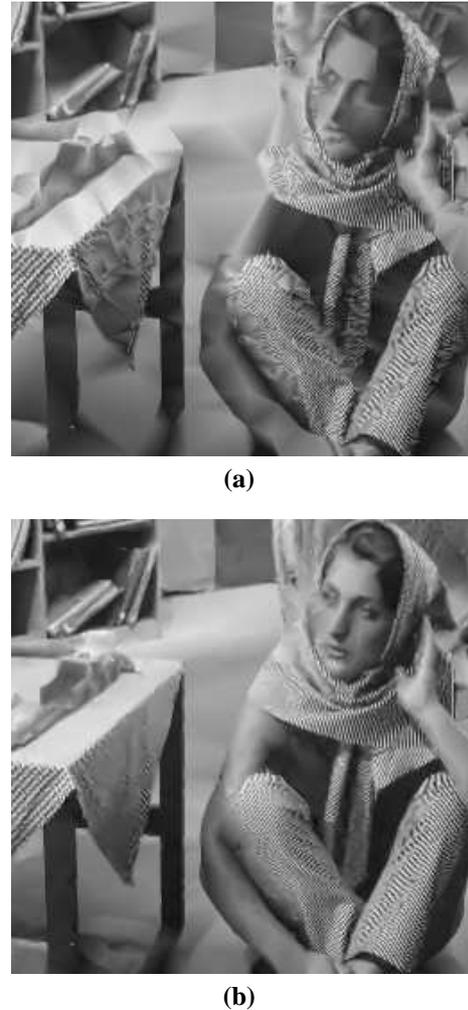


Figure 4: Barbara . Reconstruction from  $n = 7336$  most significant pixels (a) with  $L^\infty$ -norm and without LSA; (b) with  $L^2$ -norm and with LSA.

### 3. CODING SCHEME

This section briefly explains the coding of the most significant pixels, which is also subject of the previous paper [3]. First note that any coding scheme requires a non-ambiguous *decoding rule* which enables the receiver to uniquely reconstruct the image. As already discussed in the previous section, this can be accomplished by using any set  $X_n$  of  $n$  most significant pixels output by adaptive thinning.

The subset  $X_n$  can be considered as a set of tridimensional points  $(x_i^{(1)}, x_i^{(2)}, z_i)$ ,  $1 \leq i \leq n$ , where  $x_i^{(1)}$  and  $x_i^{(2)}$  are the integer coordinates of the point  $x_i$  and where  $z_i = \hat{f}^*(x_i^{(1)}, x_i^{(2)}) = Q(f^*(x_i^{(1)}, x_i^{(2)}))$  is a quantized value, where the luminances  $f^*(x_i^{(1)}, x_i^{(2)})$  are output by least squares approximation. We use a uniform quantization step  $q$ , so that  $Q(z) = \lceil z/q \rceil$ . As shown in Section 2, the use of Delaunay triangulations avoids the coding of any connectivity information. Only the tridimensional locations of the points in the subset  $X_n$  are required at the decoder. Furthermore, the ordering of the nodes is not needed for the reconstruction.

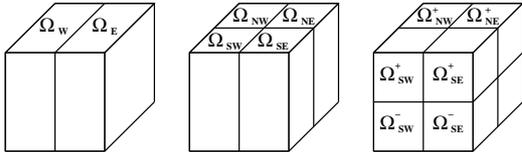


Figure 5: First three splits of the cubic domain  $\Omega$ .

We code the pixel points by performing a recursive splitting of the domain  $\Omega = [0..N] \times [0..M] \times [0..P]$ , where  $N$  and  $M$  are the dimensions of the image and  $P$  is the number of possible values for  $\hat{f}_i^*$  (typically,  $P = 2^8 - 1 = 255$  for unquantized data, but  $P = 255/q$  when the quantization step is  $q$ ).

At each step, we split a non-empty domain  $\omega$ , initially  $\omega = \Omega$ , into two subdomains  $\omega_1$  and  $\omega_2$  of equal size. If  $m_\omega$  denotes the number of most-significant pixels in the domain  $\omega$ , then we have  $m_\omega = m_{\omega_1} + m_{\omega_2}$ . Thus only one of the two numbers, say  $m_{\omega_1}$ , is added to the bitstream. At the decoder, the number  $m_{\omega_2}$  will be deduced from  $m_\omega$  and  $m_{\omega_1}$ . Each number is coded by the minimal number of required bits. For instance, since  $0 \leq m_{\omega_1} \leq m_\omega$ , the number  $m_{\omega_1}$  is coded by  $\lceil \log_2(m_\omega + 1) \rceil$  bits. The splitting of the subdomains is performed recursively until the points are exactly localized. For the purpose of illustration, the first three splits of the cubic domain  $\Omega$  are shown in Figure 5. For further details on this particular coding scheme, we refer to [3].

### 4. NUMERICAL RESULTS

We have implemented the compression scheme proposed in this paper. In this section, numerical examples are used in order to evaluate the performance of our method. To this end, we compare our compression scheme with the wavelet-based compression scheme SPIHT [6]. We evaluate the reconstruction quality of the decoded image  $\hat{I}$  by

using the *Peak Signal to Noise Ratio* (PSNR),

$$\text{PSNR} = 10 \times \log_{10} \left( \frac{255^2}{\text{MSE}} \right),$$

given in dB, where MSE denotes the *Mean Square Error*

$$\text{MSE} = \frac{1}{N \times M} \sum_{i,j} |I(i,j) - \hat{I}(i,j)|^2.$$

In our first example, we decided to use a test image called `Peppers`, shown in Figure 6 (a), whose size is 256-by-256 pixels. Note that this image contains very few textured areas. Our method provides a PSNR value of 31.13 dB (corresponding to the image in Figure 6 (b)), whereas SPIHT yields a better PSNR value of 31.65 dB (Figure 6 (c)).

A second example is shown in Figure 7. The size of the test image, called `Fruits` (shown in Figure 7 (a)), is also 256-by-256 pixels. In this test case, our method provides a PSNR value of 32.13 dB (Figure 7 (b)), whereas SPIHT yields a PSNR value of 32.77 dB (Figure 7 (c)).

Both examples show that our algorithm achieves, in contrast to SPIHT, accurate localization of sharp edges, and so it avoids spurious ringing artefacts. Although our method is slightly inferior to SPIHT in terms of its PSNR, we believe that it is quite competitive. This is supported by the good visual quality of the image reconstructions by our compression method (see Figure 6 (b) and Figure 7 (b)).

### 5. ACKNOWLEDGEMENT

The authors were partly supported by the European Union within the project MINGLE, HPRN-CT-1999-00117.

### 6. REFERENCES

- [1] Å. Björck: *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, 1996.
- [2] G.M. Davis, A. Nosratinia: Wavelet-based image coding: an overview, *Appl. Comp. Control, Signal & Circuits*, B.N. Datta (ed), Birkhauser, 1999, 205–269.
- [3] L. Demaret, A. Iske: Scattered data coding in digital image compression, *Curve and Surface Fitting: Saint-Malo 2002*, A. Cohen, J.-L. Merrien, L.L. Schumaker (eds), Nashboro Press, Brentwood, 2003, 107–117.
- [4] N. Dyn, M.S. Floater, A. Iske: Adaptive thinning for bivariate scattered data, *JCAM* **145**, 2002, 505–517.
- [5] F.P. Preparata, M.I. Shamos: *Computational Geometry*, 2nd edition, Springer, New York, 1988.
- [6] A. Said, W.A. Pearlman: A new, fast, and efficient image codec based on set partitioning in hierarchical trees, *IEEE Transactions on Circuits and Systems for Video Technology* **6**, 1996, 243–250.
- [7] D. Taubman: High performance scalable image compression with EBCOT, *IEEE Transactions on Image Processing*, July 2000, 1158–1170.
- [8] D. Taubman, M.W. Marcellin: *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer, Boston, 2002.



(a)



(b)



(c)

Figure 6: Peppers . (a) Original image; (b) compressed at 0.44 bpp by our method; (c) at 0.44 bpp by SPIHT.



(a)



(b)



(c)

Figure 7: Fruits . (a) Original image; (b) compressed at 0.57 bpp by our method; (c) at 0.57 bpp by SPIHT.