

Normal offsets for digital image compression

Ward Van Aershot^{* *}, Maarten Jansen[◇]

Adhemar Bultheel[◊]

^{*} *Dept. of Comp. Science, K.U.Leuven, Celestijnenlaan 200A,B 3001 Leuven (Belgium)*

[◇] *Dept. of Maths & Comp. Science, TU/e, 5600 MB Eindhoven (The Netherlands)*

[◊] *Dept. of Comp. Science, K.U.Leuven, Celestijnenlaan 200A,B 3001 Leuven (Belgium)*

ABSTRACT

The wavelet transform is well suited for approximation of two dimensional functions with certain smoothness characteristics. Also point singularities, e.g. texture-like structures, can be compactly represented by wavelet methods. However, when representing line singularities following a smooth curve in the domain – and should therefore be characterized by a few parameters – the number of needed wavelet coefficients rises dramatically since fine scale tensor product wavelets, catching these steep transitions, have small local support. Nonetheless, for images consisting of smoothly colored regions separated by smooth contours most of the information is comprised in line singularities (e.g. sketches). For this class of images, wavelet methods have a suboptimal approximation rate due to their inability to take advantage of the way those point singularities are placed to form up the smooth line singularity. To compensate for the shortcomings of tensor product wavelets there have already been developed several schemes like curvelets,² ridgelets,⁴ bandelets¹⁰ and so on. This paper proposes a nonlinear *normal offset* decomposition method which partitions the domain such that line singularities are approximated by piecewise curves made up of borders of the subdomains resulting from the domain partitioning. Although more general domain partitions are possible, we chose for a triangulation of the domain which approximates the contours by polylines formed by triangle edges. The nonlinearity lies in the fact that the normal offset method searches from the midpoint of the edges of a coarse mesh along the normal direction until it pierces the image. These piercing points have the property of being attracted towards steep color value transitions. As a consequence triangular edges are attracted to line up against the contours.

Keywords: *normal offsets , multiresolution, nonlinear approximation, piecewise smooth*

1. INTRODUCTION

The goal of this paper is to compactly represent the class of images consisting of smoothly gray colored areas separated by smooth contours. These images can also be seen as a piecewise smooth surface lying in the three dimensional space. This surface will be approximated by a mesh $\hat{\mathcal{M}}^\dagger$. Mostly the locations of the vertices \mathcal{V} - which define the geometry of the mesh- are represented by a list of indexed triples (x_i, y_i, z_i) . Every z_i coordinate has to be accompanied by its domain location (x_i, y_i) . However, one can also start from a coarse mesh \mathcal{M}_j and construct a new mesh \mathcal{M}_{j+1} by adding new vertices \mathcal{V}_{j+1} expressed in terms of old vertices. Normal meshes accomplish this by expressing a vertex as lying on distance from the midpoint of two old mesh points \mathcal{V}_j in a normal direction. So, only one scalar value is needed to fix a vertex's position. The meshes that can be built with such a method are called normal meshes. Normal meshes can be represented by a sequence of scalar values and as a consequence need three times less input data than the vector representation. Normal approximations of curves have already extensively been studied in [Daubechies et al.]⁸. In [Guskov et al.]⁷ normal meshes are constructed for compact representation of three dimensional surfaces and in [Jansen et al.]⁹ normal offsets are used for piecewise smooth surface approximation in the functional setting. This paper investigates the suitability of normal offsets

Ward Van Aershot is a doctoral student of the Flemish Fund for Scientific Research (FWO - Vlaanderen). This work was supported by the FWO project G.0431.05.

[†]In what follows we assume that a mesh \mathcal{M} is defined by the triple $(\mathcal{V}, \mathcal{E}, \mathcal{F})$

as a tool for image compression. Some specific adjustments will be made to meet this objective. The paper is organized as follows. Section 2 briefly describes the concept of normal offsets. Section 3 puts forward some problems that arise when extending the concept as described in Section 2 towards digital images. Section 4 discusses the approximation properties considering the class of images at interest, while Section 5 examines compression properties.

2. NORMAL OFFSETS

2.1. Concept

In the one dimensional setting the normal offset method can best be explained when compared to a linear prediction step of the lifting scheme,¹² a tool used for wavelet decompositions. Both construct a polyline interpolating several sample points. Between each two subsequent sample points, both methods will predict a new point as lying at the midpoint of a line connecting the two sample points. In most cases the real function value will differ from this prediction.

As for wavelets, the prediction step stores a detail coefficient being the difference between the predicted function value and the real function value on the same location. This detail coefficient can be seen as a vertical offset with respect to the ordinate-axis (Figure 1(a)). In contrast, the normal offset method will search for a piercing point along a ray normal to the line segment until it pierces the function (Figure 1(b)). The signed distance between the prediction point and the piercing point is kept. The old sample points are connected with the new points, forming an interpolating polyline approximation of the function.

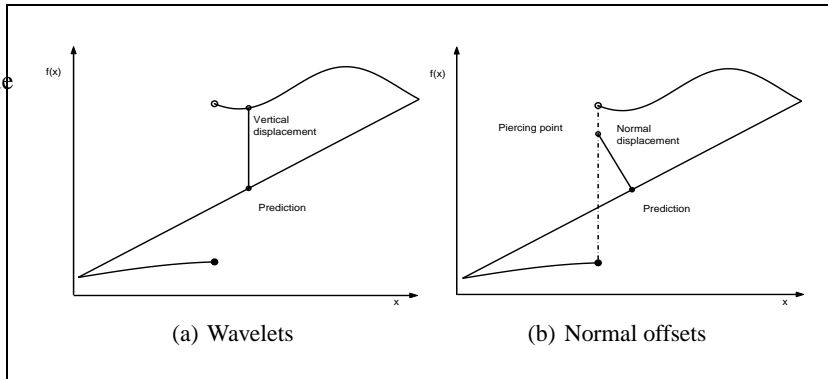


Figure 1. Conceptual comparison between wavelets and normal offsets

Note that the lifting step acts on functions $y = f(x)$ while the normal offset method considers every point as lying on a curve $c(x, y) = 0$ in the plane. Working with a geometric construction like normal offsets on a discrete function introduces its own peculiarities. A one dimensional discrete function is defined by a sequence of couples (x_i, y_i) , $i = 0 \dots M$ on the basis of which we can construct a piecewise constant function on $[x_0, x_M]$ as $f = \sum_{i=0}^M y_i \Pi(x - x_i)$, with Π the rectangle function. This function has a staircase-like shape with discontinuities at almost every knot. Therefore, the use of vertical offsets is favored since one easily finds a piercing point as the corresponding function value. When using a normal search direction one cannot expect to always pierce through the ceiling of at least one of the rectangle functions. To provide a solution, the graph of the function will be completed by vertical lines at the discontinuities similar to the dashed line in Figure 1(b). The normal line will now be searched until it pierces this connected curve c , the piercing point will be placed on this line above the previously visited x_l . An extra vertical offset will be introduced to encode the distance between the piercing point and $(x_l, f(x_l))$.

3. FROM CONTINUOUS SURFACES TO DISCRETE IMAGES

For applications such as image compression, the previous scheme has to be extended to work on two dimensional functions. These functions defined on domain Ω can also be seen as surfaces lying in a three dimensional space. The surface is now being approximated by an interpolating triangular mesh. When extending normal polylines to normal meshes extra degrees of freedom have to be filled in. A normal direction can be defined in several ways, with respect to different hyperplanes. Another problem arises from the use of a discrete set of domain points making up the regular grid, where special care should be taken in edge-refinement methods (rasterization). The next paragraphs are devoted to matters concerning the definition of the normal direction in the two dimensional setting and aspects related to discrete edge refinement.

3.1. Normal direction

First of all we assume a base mesh \mathcal{M}_0 already at our disposal. From this mesh we want to build a mesh containing more detail about the image surface. The projection of the straight line graph induced by the edges and vertices $(\mathcal{E}, \mathcal{V})$ of \mathcal{M} , defines a partition $\Delta = \{\mathcal{T}\}$ on Ω . Jansen et al. defined the normal direction as going through the midpoint of an edge and standing normal to a surface fitting its four neighbor vertices. Another approach could be shooting a ray normal to mesh triangular face \mathcal{F} while going through its center. The main drawback of such methods is that the projection on the domain of the piercing points is not constrained to be local, hereby destroying previous existing topologies. By local we mean that the piercing points lie in the same subdomain \mathcal{T} as the vertices they depend on.

Instead of global topological refinements we opt for a local refinement strategy. As such, the different subdomains keep their independence during further decomposition. Having a local refinement scheme subsequent partitions will be nested, i.e. $\Delta_j \subset \Delta_{j+1} \dots$. This assures that, when shrinking (or thresholding) some normal offsets – thereby changing the location of the mesh points \mathcal{V} and their projection– the topology will be unaffected. This makes the method topologically more stable. The locality of such method could be forced by letting the search for a piercing point not exceed the boundary of its originating domain triangle \mathcal{T} . Or, it could be developed such that piercing points are always found within the subdomain the normal ray was shot by restricting the ray to lie in a plane perpendicular to the xy -plane. This results in an edge refinement procedure and a quad-tree structured subdivision scheme. It is this definition of a normal direction that we will use from now on.

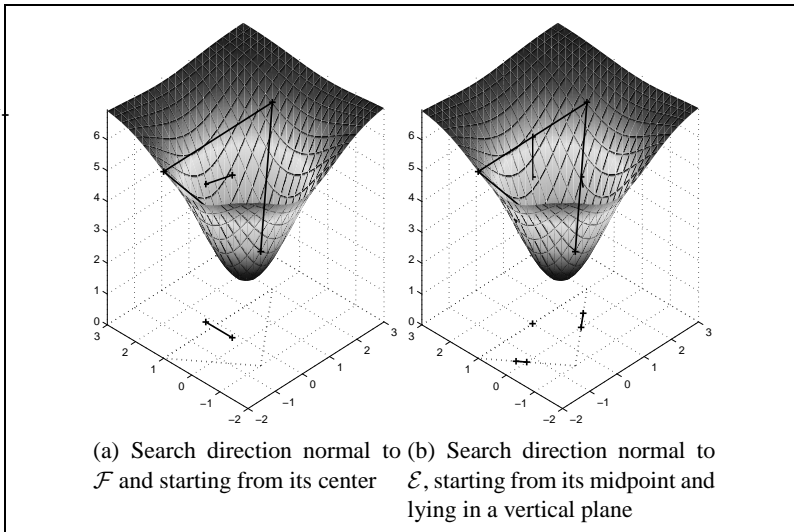


Figure 2. (Left) Piercing points are found outside the triangles domain. (Right) Piercing points are found with their location on the domain boundary

3.2. Discrete edge refinement

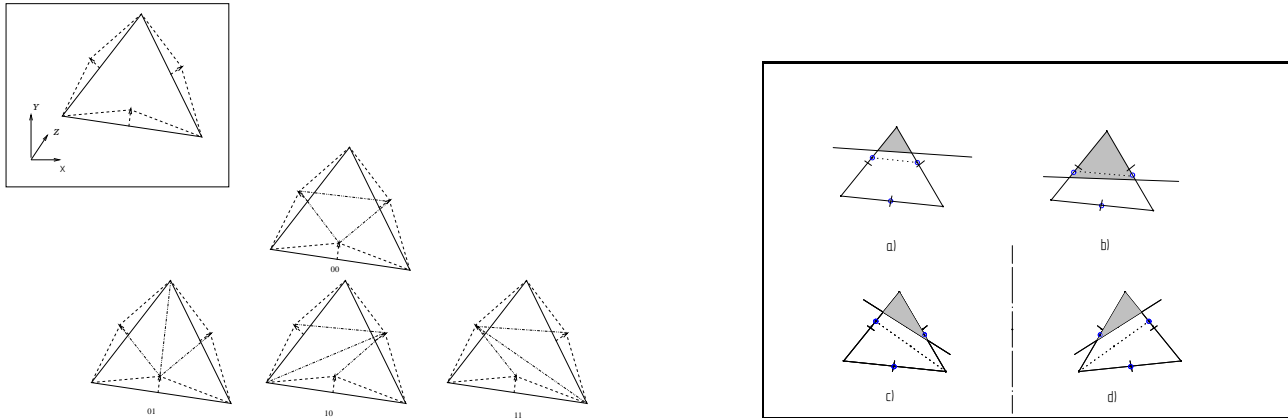
The restriction of the normal direction to lie in a vertical plane results in an edge refinement method. The edges of each $\mathcal{T} \in \Omega$ are recursively subdivided. All works fine when applied to a continuous domain $\Omega = [0, 1]^2$ on which $f(x, y)$ is defined. The prediction point lies exactly above the midpoint of the edge. In the digital setting, the point lying exactly in between two end points of a triangle edge will almost never coincide with some pixel center. A roundoff to the nearest pixel center and a connection with both end points will result in a penetration of other triangles or in pixel locations being missed out. To overcome these problems domain edges are being defined as a set of pixel locations. And its subdivided parts are forced to use a subset of this set. The midpoint (i.e. the location of the prediction point) is simply expressed as an index halfway between begin and end point of the edge. These pixel locations are the result of a rasterization process. In this paper we used the Bresenham algorithm¹ which efficiently implements such a rasterization.

4. APPROXIMATION WITH NORMAL OFFSETS

4.1. Comparison with wavelets

The strength of the compression characteristics of a method is mostly measured by its n -term approximation rate with respect to a certain space of functions. In approximation theory, it is well known that certain function norms (like Besov norms) are equivalent to a sequence norm applied to the wavelet coefficients. It is also known that functions, which can be approximated by a nonlinear *wavelet* approximation method with a certain rate, are characterized as lying in an interpolation space between an L_p space and a Besov space (we refer to Devore³ for a deeper study of nonlinear approximation).

In what follows we call the Horizon class \mathcal{H} the class of images consisting of uniformly colored region separated by a smooth contour. Since the Horizon class \mathcal{H} consists of images $f(x, y)$ that also belong to $\mathcal{B}_{2,\infty}^{1/2}$, *normal offset* approximation outperforms the best n -term *wavelet* approximation rate which is $O(n^{-1/2})$.⁶ The reason for this is that only a small subset of images belonging to certain smoothness spaces represent realistic images. Furthermore only a small subset of the images belonging to certain smoothness spaces also belong to \mathcal{H} . From the connection between wavelet coefficients and Besov norms it can be seen that the order in which the coefficients in each resolution level appear does not matter. In each resolution level, swapping the position of the wavelet coefficients –and thus destroying artifacts like edges– does not change the smoothness characteristics of the image after reconstruction. Also sign reversal of wavelet coefficients–because wavelets form unconditional bases for Besov spaces– does not affect the Besov norm. However, the reconstructed image is unlikely to resemble a realistic image. Although smooth edges could be represented (or at least approximated) by few parameters, wavelets do not take advantage of this redundancy in the domain. Where wavelets are well suited to catch point singularities they fail when it comes to line singularities. It should be clear that most likely there exist methods that exploit the inherent properties of images $\in \mathcal{H}$ to achieve a better approximation rate.



(a) (Top left) Three piercing points together with the old mesh point \mathcal{V}_j are shown. No edges are yet defined. (Bottom right) Four possible interconnections between the set of piercing points and vertices \mathcal{V}_j .

(b) Piercing points are always situated between the discontinuity and the midpoint of the edge. The gray area represents Ω where $f(x, y)_{\Omega} \equiv 1$. The dotted line must be part of the chosen interconnection since they subdivide with the maximum number of perfect approximated subtriangles.

Figure 3.

Normal offsets try to approximate the line singularities by approximating the contour $c(x, y) = 0$ by a polyline, consisting of triangle edges. When constructing a higher detailed mesh the new piercing points have to be connected by edges and a new partition $\Delta_{j+1} \subset \Delta_j$ is defined. If one refines all triangles as the first one shown at the bottom of Figure 3(a) no additional information about the mesh topology has to be stored. The compressed data only consists of geometrical information of which the topology is fixed in advance. Unfortunately, with this scheme we cannot expect that triangle edges, making up the polyline, take on a smooth path along the contour, but rather a tooth-shaped one with many segments crossing the contour. We therefore introduce some extra topological parameters – with an increase of the storage cost for the reconstruction of intermediate normal meshes – as long as it pays off at the level of compression when those meshes are truncated. The four possible interconnections of piercing points and previous vertices are depicted in Figure 3(a).

For images of \mathcal{H} we propose to take the interconnection that has at least one edge parallel to the xy -plane (and there should be at least one such interconnection). If there are several possibilities left, choose the one which minimizes the L_p error. In case of more minima, we choose the one which partitions its parent triangle such that the smallest angle is maximal (regularization). Algorithm 1 gives an overview of the full normal offset method for images $\in \mathcal{H}$. Since testing if there exist some edges that have equal function values is less computational demanding than calculating an L_p -error,

line 13 is tailor made for images of the Horizon class, which causes a speedup in execution time.

Algorithm 1 Normal Offset algorithm for Images $\in \mathcal{H}$

Require: a base mesh \mathcal{M}_0 consisting of vertices \mathcal{V} and edges \mathcal{E} .

```

1: Express each  $\mathcal{E} \in \mathcal{M}_0$  by the sequence  $\{l_i\}, i = 0, 1, -1, 2, -2, 3, -3, \dots, \frac{|E|}{2}$   $l_i \in \mathbb{N} \times \mathbb{N}$ , as the output of the
   Bresenham $(l_{-\frac{|E|}{2}}, l_{\frac{|E|}{2}})$  algorithm.
2: for  $j=1 \dots J$  do
3:   for each triangle  $\mathcal{F}_k$  in  $\mathcal{M}_j$  do
4:     for each  $\mathcal{E}_{m,k} \in \partial\mathcal{F}_k$ : do
5:       calculate the normal line  $n(x, y) = z$ .
6:        $d_0 := n(l_{i(0)}) - f(l_{i(0)})$ .
7:       repeat
8:          $k := k + 1$ 
9:          $d_k := n(l_{i(k)}) - f(l_{i(k)})$ 
10:      until  $d_i$  changes sign
11:       $p_{m,k} := (l_{i_{k-1}}, n(l_{i_{k-1}}))$ 
12:    end for
13:     $\mathcal{G} := \{g\}$ , the set of interconnections as depicted in Figure 3(a).
14:    Delete  $g$  of which the number of edges parallel to the  $xy$ -plane is not maximal
15:    if  $\#\mathcal{G} \neq 1$  then
16:      Delete  $g$  of which  $\epsilon_{L_p}$  is not minimal
17:      if  $\#\mathcal{G} \neq 1$  then
18:        Delete  $g$  of which the minimal angle is not maximal
19:      end if
20:    end if
21:     $g^* := g$ 
22:    for all edges  $\mathcal{E}^* \in g^*$  do
23:      if both end points are newly found piercing points then
24:        Assign to edge the set of pixels by running the Bresenham algorithm.
25:      else
26:        Assign locations in accordance with there parents set of locations. (this can be done by an index).
27:      end if
28:    end for
29:  end for
30: end for

```

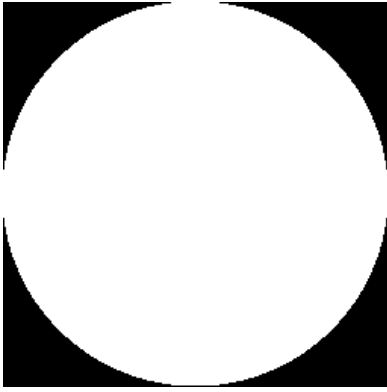
After this decomposition we have a tree structured hierarchical triangulation at our disposal. From this tree we select those normal offsets which produce a triangular mesh $\hat{\mathcal{M}}$ with the smallest difference with respect to $f(x, y)$. This is done by a pruning algorithm which prunes all descendants of a leaf's parent such that the introduced error is as small as possible. It can be proven that the normal offset method as described in Algorithm 1 has an n -term approximation rate of $\epsilon_{L_1} = O(n^{-1})$. Figure 4 shows the output of Algorithm 1 followed by an n -terms a pruning algorithm applied to the digital image of Figure 4(a).

5. COMPRESSION

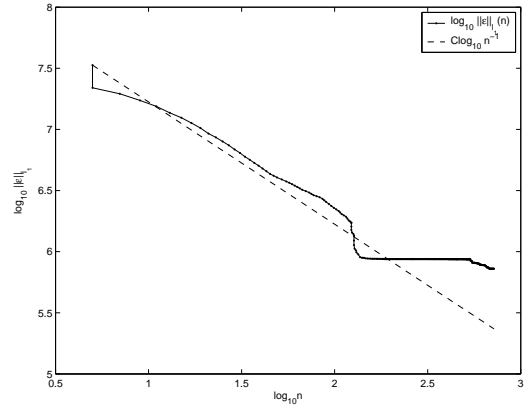
5.1. Lossless Compression

To reduce the amount of data, all coefficients should be stored with the smallest possible number of bits. This can be reached when we know something about the probability of coefficient values in terms of a PDF (probability density function) in advance. Entropy encoders, like Huffman encoders, map each value of a finite alphabet \mathcal{A} to a string of bits (codeword), $(C : \mathcal{A} \rightarrow \{0, 1\}^r)$, such that the expected number of bits is minimized.

Because digital images are represented as $2\frac{1}{2}D$ surfaces (surfaces lying in a 3 dimensional space represented by a 2 dimensional function) one can use the xy -plane as projection plane. The trick is not to store the signed length of the



(a)



(b)

Figure 4. Subfigure (a) depicts the original image $\in \mathcal{H}$ with $c(x, y) = x^2 + y^2 - R^2 \equiv 0$. Subfigure (b) shows the approximation rate of the normal offset method applied to Figure 4(a).

normal offset $n_{j,k}$ but rather the signed length $i(n_{j,k})$ of its projection on the xy -plane. The absolute value of the latter can be seen as the number of cubes to be traversed along the rasterized line starting from the midpoint of the edge till the piercing point is reached. The sign denotes in which direction to travel.

It can easily be seen that $i(n_{j,k})$ is bounded by half the rasterized edge length, and that these bounds will monotonically decrease as the resolution level j rises. However the magnitude of the normal indices themselves need not to decrease. These normal indices can be seen as *random* variables \mathbf{X} with values coming from an alphabet \mathcal{A} of size N .

$\mathcal{A} = \{0, \dots, N-1\}$, with $N = \left\lceil \frac{|\mathcal{E}_{j-1,k}^{XY}|}{2} \right\rceil$. Assume we can model this source of information by means of a Markov process. Then, every symbol \mathbf{X}_i has a probability p_i . The amount of information contained in a symbol sequence is measured by the entropy \mathcal{S} :

$$\mathcal{S} = - \sum_{i=1}^N p_i \log_2(p_i)$$

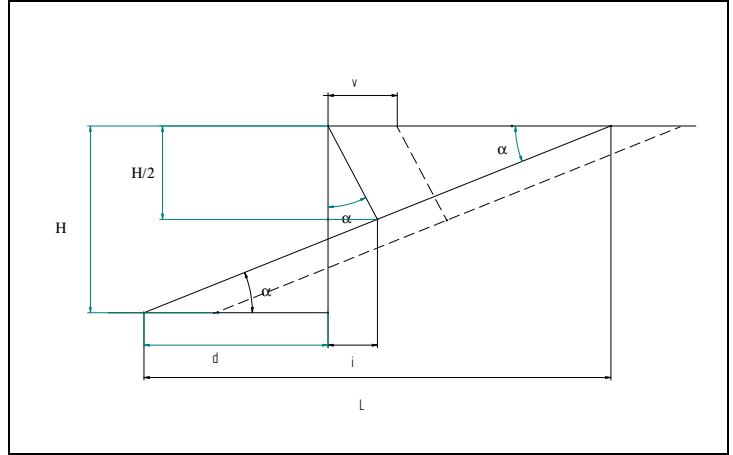


Figure 5. Crosscut of an image $\in \mathcal{H}$ containing a contour.

For a comprehensive reading about information theory we refer to the paper of C.E. Shannon.¹¹ This entropy represents the minimal number of bits per symbol, under the assumption that the sequence of symbols is very large. If we would not reckon with the statistical properties of the information source and represent all N possibilities with $\log_2 N$ bits and an equal probability, \mathcal{S} will reach its maximum value $\mathcal{S}_{\max} = \log_2 N$.

We will now give an a priori PDF of these normal indices when given two edge points, considering $f_{\Omega}(x, y) \in \mathcal{H}$, where $\Omega = [0, 1] \times [0, 1] \subset \mathbb{R} \times \mathbb{R}$. If the distance d as depicted in Figure 5 is uniformly distributed on the interval $[0, L]$

and $H < L$ then the distribution of $i_{n_j,k}$ – the orthogonal projection of the normal offsets on the xy -plane – will be:

$$p(i_{n_j,k}) = \begin{cases} \frac{1}{L} & -\frac{H \tan \alpha}{2} < i_{n_j,k} < \frac{H \tan \alpha}{2} \\ \frac{1}{2} \left(1 - \left(\frac{H}{L}\right)^2\right) & i_{n_j,k} = \pm \frac{H \tan \alpha}{2} \\ 0 & |i_{n_j,k}| > \frac{H \tan \alpha}{2} \end{cases}$$

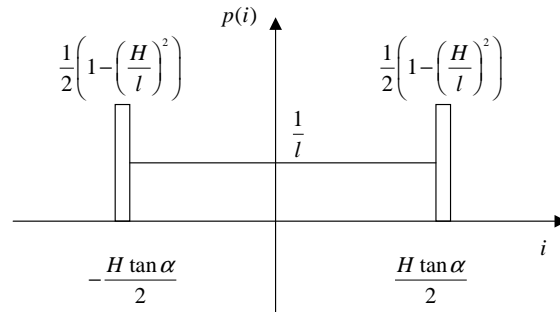


Figure 6. The above picture represents the PDF if the distance d has a uniform distribution on $[0, L]$. The smaller the height H compared to L , the larger the probability at $i = \pm H \frac{\tan \alpha}{2}$ (large probability to pierce through the one of the horizontal regions) and the lower the uniform distributed probability between $[-H \frac{\tan \alpha}{2}, H \frac{\tan \alpha}{2}]$

In Figure 7 the value of S is plotted for different values of L . The Figure also illustrates that when L rises the profit due to the a priori known distribution gets larger for a larger part of H -values.

The entropy encoder generates a reduction of data without loss of information. In view of rate distortion theory, this agrees with a reduction of the distortion even at the very beginning of the R/D-curve. This lossless compression is a full gain to the bit rate. Table 8(a) shows results for several test images. Figure 8(b) schematically represents on which data the decoder and encoder can lay their hands on to produce a table which contains a codeword for each possible value (and visa versa).

As a last step, all symbols can then be processed by an arithmetic encoder. This encoder maps all the data –considering the frequency of the codewords – into one string of bits (representing a fractional value between 0 and 1) uniquely representing the original data.⁵

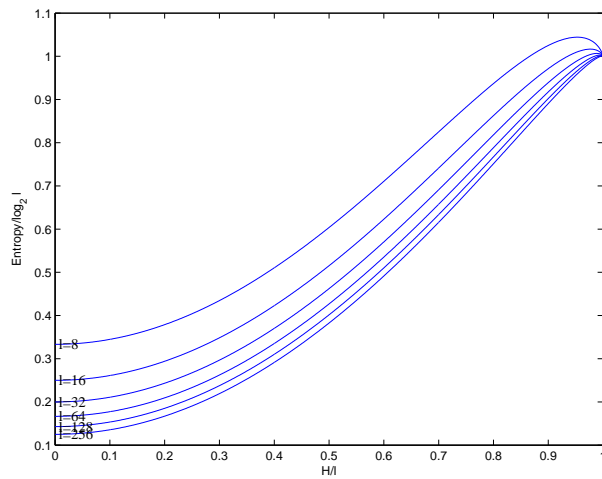
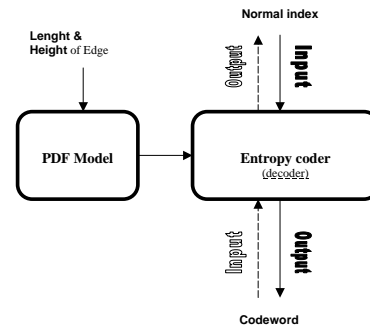


Figure 7. $S \in [0, 1]$. For some values of L the entropy S reaches values larger than $\log_2 L$. This is because we used a continuous set of values instead of finite set \mathcal{A} . For larger L , this continuous expression approximates the discrete one. When $H/L = 1$ the PDF is a uniform distribution over $[-L, L]$ and S takes on its maximal value.

Figure $f(x, y)$	$f(x, y) \in \mathcal{H}$?	compression ratio
circle (Figure 4(a))	yes	1:2
block (Figure 9(a))	yes	1:2
Lena	no	5:7

(a) This table shows some compression ratios on the normal indices for several test images.



(b) Schematic illustration of the entropy encoder and decoder. On the basis of the data of the edge, the normal indices can be mapped to codewords and visa versa.

Figure 8.

6. RESULTS AND CONCLUSIONS

We have tested our normal offset algorithm (Algorithm 1) on different kinds of test images, i.e. an Horizon class image, an image with smoothly colored regions separated by smooth contours and a realistic picture better known as ‘Lena’. Note that the Horizon class is a subset of the class of images with smoothly colored regions and smooth contours which in its turn is a subset of the more general class of realistic pictures. For all images, the algorithm started with a base mesh \mathcal{M}_0 consisting of two triangles built from the four corners of the domain. In Figure 9 the normal offset decomposition of the top image $\in \mathcal{H}$ is subjected to an n -terms selection algorithm. We see that the triangulation needs few coefficients to align along the straight line discontinuity. In Figure 10 an Horizon class image with a curved contour is shown together with several n -terms approximations. If we compare this result with Figure 11 where the same image is approximated by a biorthogonal $(2, 2)$ wavelet method with the same number of terms, we see that the contours are far more blocky. At last, the output for the ‘Lena’ picture is shown in Figure 13. In Figure 13(a) a pruned domain triangulation is shown. Even without color information we can distinguish Lena’s shoulder, face contours, border of the mirror and parts of her hat. These are typically areas which own the properties suited for our normal offset algorithm. Figures 13(b)-13(d) represent an intermediate mesh at resolution level $j = 6$. From Figure 13(d) we notice that particular areas are well approximated while sharp curved areas, like Lena’s nose and eyes, show deviations from the original image.

Instead of using such a simple base mesh \mathcal{M}_0 as used in our experiments, further research will involve base meshes which already represent the basic geometry of the image. This will demand more data to store the initial mesh –as it will be represented by a vector representation rather than a scalar one – but will lead to better quality approximations afterward. In the hope artifacts as seen in Figure 10 will disappear.

As we see in Figure 12, once the pruning algorithm prunes a triangle that had a leaf along the contour in the vertical direction, a great L_2 error is introduced. The overall rate is the same as for the Horizon class images.

For images with smoothly colored areas and images like ‘Lena’, algorithm 1 will be expanded to use other norms for interconnection rather than solely an L_2 distance norm. Curvature norms will be combined with distance norms – like the L_2 norm– which construct interconnections that resemble a similar shape as the above surface. If the shapes matches the surface for some parts of an intermediate mesh, next refinements will have it easier to adapt towards the image surface. As a consequence we could expect a better approximation rate towards these images similar to that of images of the Horizon class.

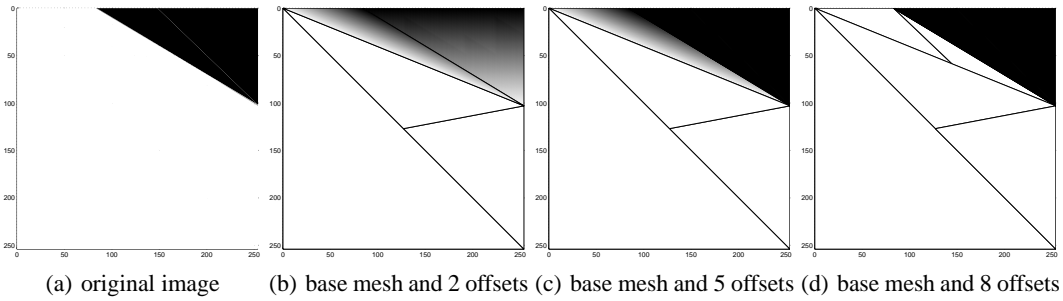


Figure 9. Picture (b)-(d) consist of only n (normal + vertical offsets). They are produced by an implementation of Algorithm1 followed by a pruning scheme on Picture(a).

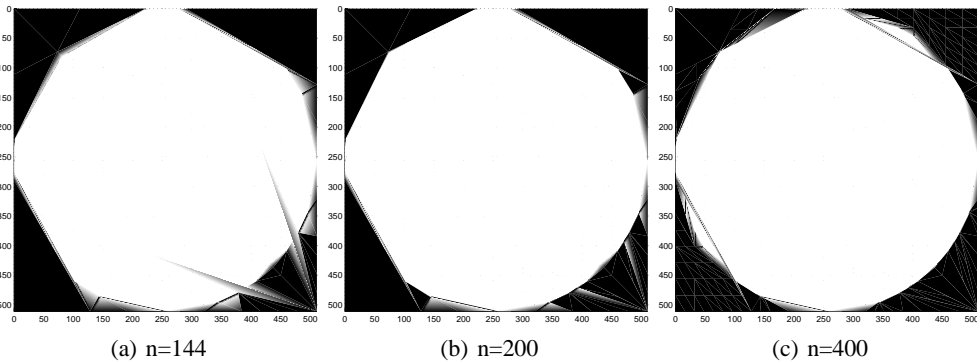


Figure 10. Compressed images of the original image in Figure 4(a) consisting of only n (normal + vertical offsets). They are produced by an implementation of Algorithm1 followed by a pruning scheme.

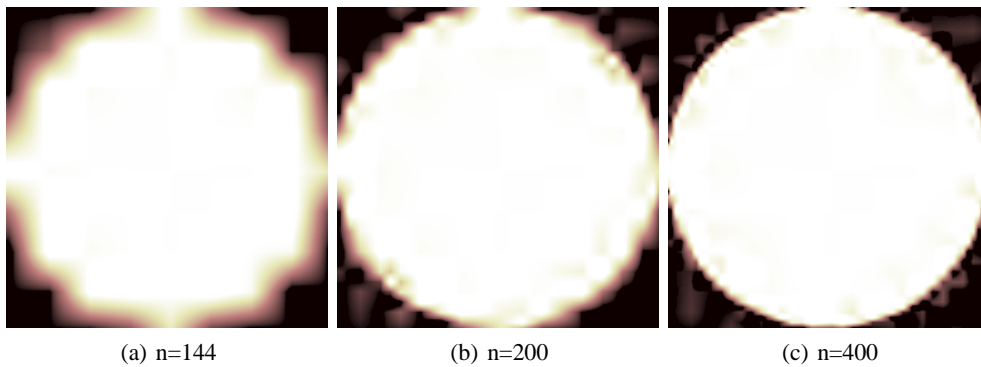


Figure 11. Compressed images of Figure 4(a) where only the n most significant wavelet coefficients are kept. We used a biorthogonal (2,2) wavelet decomposition scheme, which uses a linear prediction step corresponding to the prediction used in the normal offset algorithm.

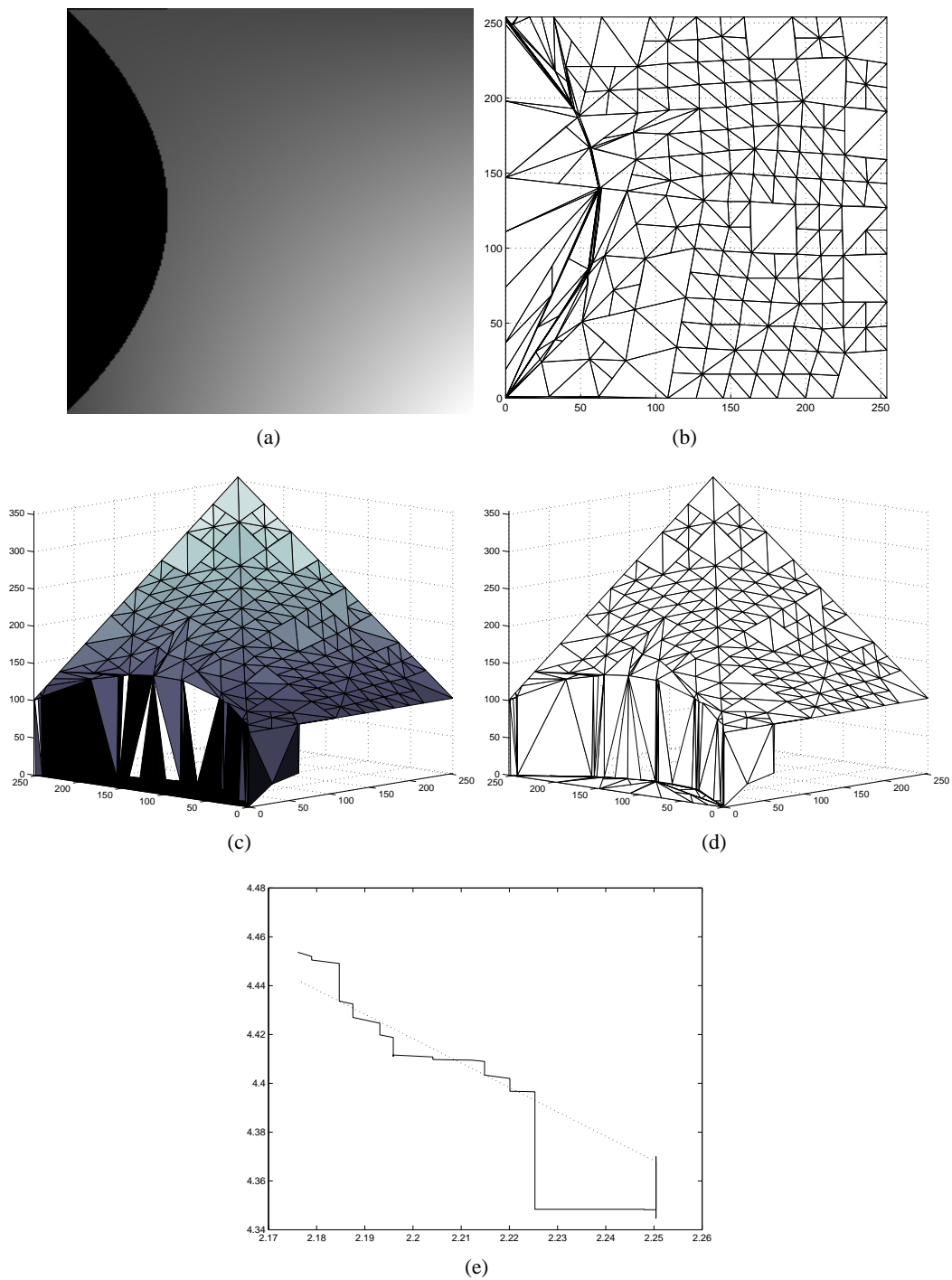
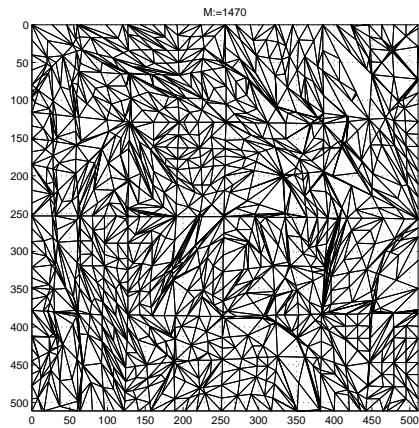
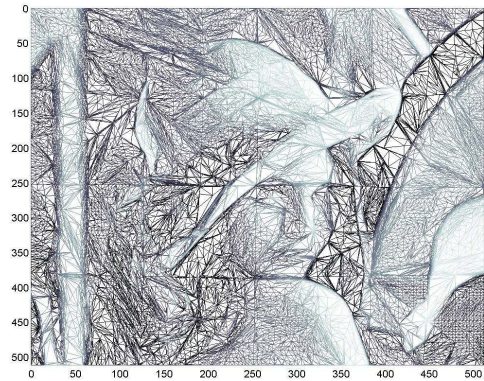


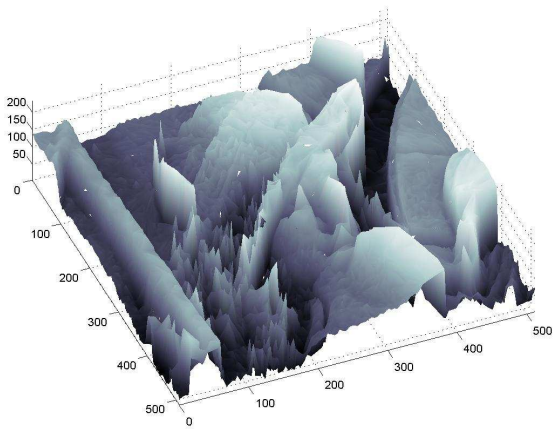
Figure 12. Figure 12(a) shows the original image. Figure 12(b) shows the domain partitioning after the normal offset decomposition. Figure 12(c) and Figure 12(d) show the approximations of the surface by a mesh containing a certain number of normal offsets, $n = 240$. As we see in Figure 12(e), once the pruning algorithm prunes a triangle that had a leaf along the edge in the vertical direction, a great L_2 error is introduced. This results in some transition phenomena in the n -terms approximation curve and sudden fast drop downs of the approximation error. But we see from Figure 12(e) that the output after loosing track comes back to the theoretical approximation line.



(a) A pruned domain triangulation of *Lena*.



(b) Mesh representation of \mathcal{M}_6 from which one can see the dense triangulation near contours.



(c) The image surface is well approximated when it behaves smooth and has discontinuities along smooth curves, which is clearly the case for *Lena*'s shoulder and the border of the mirror.



(d) A colored version of Figure 13(b), where the area around shoulder, hat, mirror border and face have a sharp representation.

Figure 13. *Lena*

REFERENCES

1. J. E. Bresenham. Algorithm for computer control of a digital plotter. pages 1–6, 1998.
2. E. J. Candès and D. L. Donoho. Curvelets - a surprisingly effective nonadaptive representation for objects with edges. Technical report, Department of Statistics, Stanford University, 2000.
3. R. A. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.
4. D. L. Donoho. Orthonormal ridgelets and linear singularities. *SIAM J. Math. Anal.*, 31:1062–1099, 2000.
5. Jr. G. G. Langdon. An introduction to arithmetic coding. *IBM Journal of Research and Development*, 28(2):135, 1985.
6. C. Sinan Güntürk. How much should we rely on besov spaces as a framework for the mathematical study of images? In *Wavelet and Applications Workshop (WAW'98)*, 1998.
7. I. Guskov, K. Vidimee, W. Sweldens, and P. Schröder. Normal meshes. In *SIGGRAPH 2000 Conference Proceedings*, 2000.
8. O. Runborg I. Daubechies and W. Sweldens. Normal multiresolution approximation of curves. *Constructive Approximation* :20, pages pp.399–463, 2004.
9. M. Jansen, R. Baraniuk, and S. Lavu. Multiscale approximation of piecewise smooth two-dimensional functions using normal triangulated meshes. *Appl. Comp. Harm. Anal.*, 2005.
10. E. Le Pennec and S. Mallat. Sparse geometrical image representations with bandelets. *submitted*, 2003.
11. C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:pp. 379–423 and 623–656, July and October, 1948.
12. W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2):186–200, 1996.