Digital normal meshes in image compression

Ward Van Aerschot
Maarten Jansen
Adhemar Bultheel

Report TW446, January 2006



Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Digital normal meshes in image compression

Ward Van Aerschot
Maarten Jansen
Adhemar Bultheel

Report TW446, Januari 2006

Department of Computer Science, K.U.Leuven

Abstract

This paper concerns compression of digital images consisting of smoothly colored regions separated from each other by smooth contours. Wavelet transforms combined with a nonlinear thresholding step are optimal when catching point singularities, however, when trying to capture smooth line singularities in a parsimonious manner they perform suboptimal. We propose a geometrical method to deal with these inherent deficiencies of wavelets. The proposed method using normal offsets transforms the input image nonlinearly using only a series of scalar values. Special attention is given on how to adapt the concept of normal offsets towards the digital setting. Moreover, a local adaptive triangulation is issued such that the method is able to sparsely represent higher dimensional singularities, i.e., line singularities. The adaptive triangulation encourages the triangle edges to settle themselves parallel with respect to the contour. At the same time the locality of the triangulation method induces hierarchical mesh structures which are well suited for compression methods.

Keywords : adaptive, normal mesh, normal offset, multiscale triangulation, hierarchical mesh, image, compression.
AMS(MOS) Classification : Primary : 94A08, Secondary : 68U10, 49M27.

1 Introduction

Image compression methods aim at representing –approximations of– original images with as few bits as possible while controlling the quality of these representations. It is thereby not surprising that this topic is closely related to the field of approximation theory, where methods are constructed to optimally approximate certain functions using only n terms; optimally with respect to the rate at which the error norm ϵ_n tends to zero as n goes to infinity. Most often one tries to bound the error norm ϵ_n by the sequence $n^{-\alpha}$, where α is called the rate of approximation. The rate at which the n-terms approximation converges depends both on the method applied and on the characteristics of the functions to be approximated. All functions, having the same rate of approximation α –under the assumption that those approximations are obtained by the same method – are collected in an approximation space $\mathcal{A}^\alpha$. For some methods these approximation spaces can be characterized by smoothness spaces. This means that, once the characteristics, e.g. regularity, of a function are known we know its corresponding approximation rate. It is well-known that, for example, approximation spaces $\mathcal{A}^\alpha$ of nonlinear n-terms wavelet approximation methods can be characterized by smoothness spaces, i.e. Besov spaces.

This paper describes a method for approximating a class of target functions, consisting of *smooth regions* separated by *smooth contours*. These target functions all possess a certain regularity s, despite their discontinuities. Such piecewise regularity can be expressed in terms of Besov spaces $\mathcal{B}^s_{p,p}$ [1]. All spaces $\mathcal{B}^s_{p,p}$ contain discontinuous functions for every value of s. More specifically, the functions under consideration all belong to Besov space $\mathcal{B}^s_{p,p}$, satisfying $\frac{1}{p} = \frac{1}{2} + \frac{s}{d}$, with regularity parameter $0 < s \leq \frac{1}{2}$. This imposes a constraint on the maximal rate of approximation α for a nonlinear n-terms wavelet approximation method in $\mathbb{R}^d$, i.e. $0 < \alpha \leq s$. [2]

The previous result is not very satisfying in the sense that, when looking at the target functions of interest, most of the information lies in the smooth contours $c(x) \in \mathcal{C}^2$ which can be approximated like n^{-2}. The following example, which we will use throughout the text, will make things more clear.

Example 1.1. *Suppose we restrict the class of target functions even further and define the Horizon class $\mathcal{H}$ as the class of functions defined on $\Omega = [0,1] \times [0,1] \in \mathbb{R}^2$ having two constant regions which are separated by a smooth contour:*

$$\mathcal{H} = \left\{ f_\Omega(x,y)_{c(x)} \right\} \mid c(x) \in \mathcal{C}^2 \ \ \& \ \ f_\Omega(x,y)_{c(x)} = \begin{cases} H & c(x) > y \\ 0 & c(x) \leq y \end{cases}, H \in \mathbb{R} \,. \tag{1}$$

Those functions, being two-dimensional piecewise constant functions, belong to a Besov space $\mathcal{B}^s_{p,p}$ with regularity $s = 1/2$. This means that, using a nonlinear wavelet approximation method, the best one can do is approximate them such that the approximation error $\epsilon_n = O(n^{-1/2})$, when $p = 2$. However, the complexity of the $2D$ functions that belong to $\mathcal{H}$ is fully determined by the shape of $c(x)$. Because we restrict $c(x)$ to be smooth, i.e. $c(x) \in \mathcal{C}^2$, it can be approximated using only a small number of parameters. From approximation theory it is well known that using a nonlinear wavelet approximation with sufficient smoothness and a sufficient number of vanishing moments, the rate of approximation α of a twice differentiable one dimensional function equals 2.

From Example 1.1 we see that there is still a large gap between the theoretical optimal rate ($\alpha = 2$) and the rate that can be achieved by nonlinear wavelet approximation methods ($\alpha = 1/2$) . The reason why those wavelet approximation methods perform suboptimally on higher dimensions is that wavelets are originally developed to handle point singularities. In one dimension, of course, only point singularities exist. At each resolution level, the number of wavelet functions that come into contact with this discontinuity is bounded by a constant. Going to higher dimension also higher dimensional singularities come into play. For instance, in the two dimensional setting line singularities corresponding to $c(x)$ are to be dealt with. Wavelets defined on dyadically refined grids cannot handle those line singularities in a parsimonious manner. The number of –high frequency– wavelets needed to catch those steep transitions rise exponentially with the resolution level j, as their support decays like 2^{-2j}.

[1] A precise definition can be found in [3], but for the moment, think of $\mathcal{B}^s_{p,p}$ as the set of functions when having the sth order derivative in $\mathcal{L}_p$

[2] We refer to [3] for an elaborated study on nonlinear approximation.

In the literature several methods have been proposed to overcome the weaknesses of wavelet methods. All those methods try to catch the smooth line singularities in a more efficient manner. Some redundant linear decompositions using directional frames have been proposed like ridgelets [6], curvelets [2], contourlets [4], which nonlinear approximation performance on the target functions is quite impressive. Also several *highly* nonlinear (geometry based) approximation schemes like wedgelets [5] (using a dictionary approach) and bandelets [11] have been developed to handle line singularities. We make the difference between *mildly* nonlinear and *highly* nonlinear schemes, indicating that for the latter, the used decomposition method is data dependent while for the former the nonlinearity lies in the *selection* of the n terms (e.g. a thresholding step).

Recently another highly nonlinear approximation scheme referred to as *normal* approximation, has been investigated in the context of smooth curve approximation in the plane [9], of smooth surface approximation in $\mathbb{R}^3$ [8] and some analysis for the setting presented here (i.e. in the presence of line discontinuities) has been done by Jansen et. al in [10].

The paper is organized as follows. Section 3 introduces the concept of normal offsets in the continuous setting (Subsection 3.1). Subsections 3.3 and 3.4 accommodate the concept of normal offsets towards the target functions described in Subsection 3.2. Section 4 discusses the modifications that have to be made to adapt the normal offsets procedure to the discrete setting. Section 5 introduces both lossless and lossy compression properties of the proposed method. Section 6 concludes by showing how the proposed method outperforms the nonlinear wavelet approximation on this class of images.

2 Contributions

The contribution of this paper is threefold: First we use an adaptive topology algorithm to be discussed in Section 3.4.2 in the continuous setting and again in Section 5.2 for the discrete setting; Second, the application of normal offsets to *digital* images which is the main topic of Section 4; Third, the application of compression.

As there have been written many papers with regard to smoothly varying surfaces in the context of normal approximation, only [10] considers the behavior of piercing points in the presence of discontinuities. Roughly speaking, these piercing points are points of intersection of a normal to a coarse approximation of the function and the function. A precise definition will be given in Section 3. The authors show that for the one dimensional setting piercing points are attracted towards steep function transitions.

As a major improvement over their algorithm, our method exploits this one dimensional property by *constructing* edges (two dimensional features) in such a way that they are attracted towards the line discontinuities in a *tangential* manner. This means that certain triangular edges are encouraged to settle themselves parallel with respect to the contour, instead of crosscutting it. This is depicted in Figure 1. As such, several of these edges will act as a piecewise polyline approximation of the contour. This is *essential* for the improvement of the rate of approximation. Moreover, the proposed algorithm also naturally adapts the definition of a normal ray to the functional setting in such a way that our method generates hierarchical triangulations (in contrast with normal rays as defined in [8],[10]). Hierarchical triangulations can be represented by tree structures which are better suited for compression than triangulations living on different resolution levels without parent-child dependencies.

3 Normal Offsets

3.1 Concept

The concept of the normal offset method can best be illustrated in the one dimensional setting, i.e. functions $f : [0, 1] \mapsto \mathbb{R}$. Let us compare it with a linear prediction step of the lifting scheme [13], as depicted in Figure 2. In a first step, both methods construct a polyline connecting the consecutive sample points $p_{j,k} = \begin{bmatrix} x_{j,k} & y_{j,k} \end{bmatrix}^T$. This polyline is a coarse approximation of the function. To get a finer approximation new points are added to

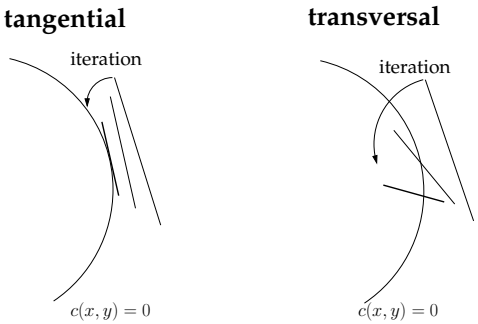


Figure 1: *The figure (left) illustrates what is meant by a* tangential *attraction of two dimensional line structures, e.g. triangular edges, towards the contour. This in contrast with a convergence of points towards the contours but with their corresponding edges crossing the contour (right).*

refine the polyline. Both methods will predict a new point as lying in the middle of the line connecting the two adjacent sample points, i.e. $\mathcal{P}(p_{j,k}, p_{j,k+1}) = p^*_{j+1,2k+1} = 1/2\,[p_{j,k} + p_{j,k+1}]$. In most cases this prediction point will not coincide with a point $(x, f(x))$, i.e. $y^*_{j+1,2k+1} \neq f(x^*_{j+1,2k+1})$. For reconstruction purposes these discrepancies are stored in displacement maps, i.e. additional data denoting the difference between a coarse level approximation and the finer level approximation. This is where both methods differ conceptually. As for wavelets, the prediction step stores a detail coefficient $d_{j+1,2k+1}$ being the difference between the predicted function value and the real function value at the same location, i.e. $d_{j+1,2k+1} = f(x_{j+1,2k+1}) - y_{j+1,2k+1}$. This detail coefficient can be seen as a *vertical* offset with respect to the ordinate-axis. 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. The signed distance between the prediction point and the piercing point is kept.

The old sample points are connected with the new piercing points, forming a finer interpolating polyline approximation of the function.

Remark 3.1. *Note that the lifting step acts on functions $y = f(x)$ while the normal offset method can be applied more general to curves since it considers every point implicitly defined by $F(x, y) = 0$.*

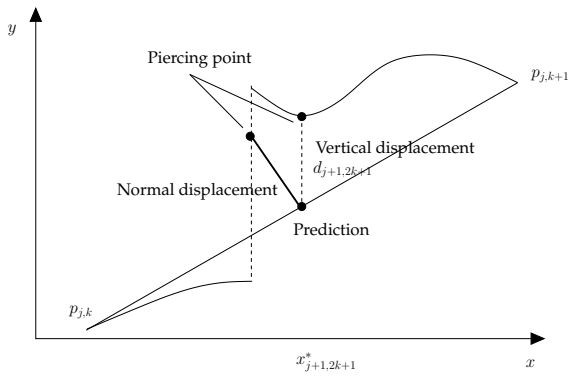


Figure 2: *Conceptual comparison between wavelets and normal offsets*

3.2 Target Functions

In this paper our interest goes to two dimensional piecewise smooth functions $f : [0,1]^2 \mapsto \mathbb{R}$ of which the abrupt function value transitions (discontinuities) lie along smooth contours. We will denote this class of functions by $\mathcal{PS}$. A special subset of $\mathcal{PS}$ already encountered in Example 1.1, is the Horizon class $\mathcal{H} \subset \mathcal{PS}$ as defined by (1). As the function $f \in \mathcal{H}$ is completely determined by $c(x)$, the goal is to rapidly approximate $c(x)$ using an *adaptive* domain partitioning method. As will be explained in Section 5.2 this shall be done by a polyline approximation imposed by an adaptive refined triangulation of Ω constructed by the normal offset method. Note that using a nonlinear tensor product wavelet approximation $c(x)$ is approximated by a piecewise constant.

3.3 2D Normal Offsets

Think of a two dimensional function $f : \mathbb{R}^2 \mapsto \mathbb{R}$ as a surface in the $3D$ space, with the domain acting as the $xy-$plane. In fact, this representation can be seen as a degenerated case of a general parametrical surface in $3D-$space. The philosophy behind normal approximation is quite natural. In the decomposition phase piercing points are found as the intersection between a normal line and the surface defined by f. Starting from a coarse interpolating mesh $\mathcal{M}_j$ of the target function f, a finer approximation $\mathcal{M}_{j+1}$ is expressed in terms of scaled vectors $\{\mathbf{n}_{j,k}\}$ lying in a predefined normal direction with respect to the coarse approximation $\mathcal{M}_j$. Only a scalar value, denoting the scaling factor with respect to the normalized normal vector has to be stored. (How adjacency of the piecing points is handled will be postponed until Section 3.4). This makes the reconstruction straightforward. As pointed out in [10] and illustrated in Section 3.1, in the one dimensional case, these piercing points are attracted towards discontinuities. The primary goal in the two dimensional setting is attracting two dimensional structures , i.e. triangle edges,in a *tangential* manner towards the line discontinuities.

Suppose we start from sequences $\{\mathbf{x}_{j,k}\}$ and $\{f(\mathbf{x}_{j,k})\}$. In what follows we shall use $f_{j,k}$ as a shorthand for $f(\mathbf{x}_{j,k})$, and use $\Delta\cdot_{j,k}$ defined as $\Delta\cdot_{j,k} := \cdot_{j,k} - \cdot_{j,k+1}$. For the sake of notational simplicity we denote

$$\mathbf{x}_{j,k} := \begin{bmatrix} x_{j,k} \\ y_{j,k} \end{bmatrix},$$

for the coordinates of the kth triangular grid point at resolution level j, and

$$p_{j,k} := \begin{bmatrix} \mathbf{x}_{j,k} \\ z_{j,k} \end{bmatrix}.$$

Next we apply the linear operator $\mathcal{P}$ such that :

$$p^*_{j+1,2k+1} = \mathcal{P}\left(p_{j,k}, p_{j,k+1}\right) = \frac{p_{j,k} + p_{j,k+1}}{2}. \tag{2}$$

and define $f^*_{j,k} := z^*_{j,k}$.

The normal offset procedure imposes that the piercing point $(\mathbf{x}_{j+1,2k}, f(\mathbf{x}_{j+1,2k}))$ satisfies:

$$(\mathbf{x}_{j+1,2k+1} - \mathbf{x}^*_{j+1,2k+1})^T \Delta\mathbf{x}_{j,k} + (f(\mathbf{x}_{j+1,2k+1}) - f^*_{j+1,2k+1})\Delta f_{j,k} = 0, \tag{3}$$

which will further be referred to as the orthogonality condition. This condition restricts the piercing points to lie in the plane orthogonal to the vector $p_{j,k+1} - p_{j,k}$ going through $p^*_{j,2k+1}$. In order to completely define a normal ray, an extra condition has to be imposed:

$$\langle((p_{j+1,2k+1} - p_{j,k}) \times (p_{j+1,2k+1} - p_{j,k+1}))^T, \mathbf{e}_z\rangle = 0 \qquad , \qquad \mathbf{e}_z := \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{4}$$

This equation restricts the piercing point lying on a plane normal to the xy-plane.

Equation (4) plays a central role in the remainder of this paper. Combined with Eq. (3) the definition of a normal offset is fundamentally different from other approaches used in [8] [10]. Geometrically this constraint

is perfectly sound in the sense that – as explained in Section 3.2– the parametrical plane coincides with the xy-plane. The constraint makes sure that $\mathbf{x}_{j+1,2k+1} := (x_{j+1,2k+1}, y_{j+1,2k+1})$ can be expressed as a convex combination in terms of $\mathbf{x}_{j,k}$ and $\mathbf{x}_{j,k+1}$. This is what makes the method an edge refining method.

The piercing point $p_{j+1,2k+1}$ has to satisfy the set of nonlinear equations:

$$p_{j+1,2k+1} := \begin{cases} (\mathbf{x}_{j+1,2k+1} - \mathbf{x}^*_{j+1,2k+1})^T \Delta \mathbf{x}_{j,k} + (f(\mathbf{x}_{j+1,2k+1}) - f^*_{j+1,2k+1}) \Delta f_{j,k} = 0 \\ \\ \langle ((p_{j+1,2k+1} - p_{j,k}) \times (p_{j+1,2k+1} - p_{j,k+1}))^T, \mathbf{e}_z \rangle = 0 \end{cases} \tag{5}$$

Definition 3.1 (Normal ray)**.** *We define a normal ray as points p satisfying :*

$$\begin{cases} (p - p^*_{j+1,2k+1})^T (p_{j,k+1} - p_{j,k}) = 0 \\ \\ \langle ((p - p_{j,k}) \times (p - p_{j,k+1}))^T, \mathbf{e}_z \rangle = 0 \end{cases} \tag{6}$$

Remark 3.2. *The piercing point $p_{j+1,2k+1}$ satisfying (5) also satisfies (6).*

Definition 3.2 (Normal vector)**.** *We define a normal vector $\vec{n}$ by*

$$\vec{n} = p_{j+1,2k+1} - p^*_{j+1,2k+1} \tag{7}$$

Corollary 3.1. *It follows from Equations (5) and (6) that only the norm –and a sign – of the vector*

$$\pm \|\vec{n}\|$$

*(*normal offset*) is sufficient for the reconstruction of $p_{j+1,2k+1}$.*

In the next iteration we declare the even points to be the old points:

$$p_{j+1,2k} := p_{j,k}$$

making the scheme interpolating.

3.4 Topology

In two dimensions adjacency is handled by graph structures, and concerning this paper, triangulations in particular. Before going into detail of how meshes are constructed from a set of piercing points, we introduce some preliminary definitions of these concepts:

Definition 3.3 (Triangulation $\mathcal{T}$)**.** *Let $V = \{\mathbf{x}_1, ...\mathbf{x}_n\}$ be a finite set of points in $\mathbb{R}^2$. A triangulation $\mathcal{T}$ of set V is a maximal plane straight-line graph $G = (V, E)$, where, E is a set of non-crossing line segments with endpoints in V.*

Definition 3.4 (Hierarchical triangulation $\mathcal{T}^{\mathcal{H}}$)**.** *An hierarchical triangulation $\mathcal{T}^{\mathcal{H}}$ of a plane domain is a triangulation $\mathcal{T}$ obtained by recursively applying a refinement procedure where each triangle is subdivided in several disjunct sub triangles.*

For a more formal definition – and an elaborated study of – hierarchical triangulations we refer to [7]. Section 3.3 explained how the new vertices $p_{j+1,k}$ are found from a coarse resolution mesh $\mathcal{M}_j$. The set of $\mathbf{x}_{j+1,k}$ coordinates, i.e. $\{\mathbf{x}_{j+1}\}$ form the vertices V_{j+1} for a triangulation $\mathcal{T}_{j+1}$ of Ω. Next to the set of vertices V_{j+1}, a triangulation needs also edges E_{j+1} to be defined, i.e. $\mathcal{T}_{j+1} = (V_{j+1}, E_{j+1})$. While adjacency in 1D is trivial, in 2D this is handled by graph structures. This section will be devoted to the construction of this set of edges E_{j+1}.

The objective is to construct a *local multi-resolution method* making use of hierarchical triangulations $\mathcal{T}^{\mathcal{H}}$ of $Hull\,(\Omega)$. An hierarchical triangulation $\mathcal{T}^{\mathcal{H}}$ (Definition 3.4) can be seen as a set of triangulations of which

each triangle of a next finer triangulation comes from a refinement of one triangle in the coarse triangulation. Such a hierarchical triangulation can be represented by a tree structure. Between triangles partial relations like parent,child, ancestor and descendant can be defined. Having an hierarchical method we can enhance details of a certain region without adapting the rest of the image.

Remark 3.3. *The construction of the proposed normal offset procedure lends itself to create hierarchical meshes in order to obtain a local multi-resolution method. Equation (4) that forces the method to be an edge refining method. Newly found piercing points $p_{j+1,2k+1}$ have their ordinates $\mathbf{x}_{j+1,2k+1}$ on the same edge $e_x(p_{j,k}, p_{j,k+1})$ the normal ray was shot from.*

In the decomposition only conforming triangulations are produced. A conforming triangulation requires that two triangles have either an edge in common or have no point in common. As we will work with a quadtree structured subdivision scheme the refinement operators *split* each triangle in four subtriangles. This splitting rule can be fixed in advance or can be made data-dependent. The latter will introduce some extra topological information to be stored for later reconstruction, but one hopes for a better approximation rate, i.e. a faster approximation of the smooth contour by a subset of E_{j+1}.

We already noted in Section 2 that *certain piercing points* $\in V$ are attracted towards steep changes in function value (a 1D concern). The splitting method uses this property, in particular for the attraction of *certain edges* $\in E$ towards the contour (a 2D concern).

Following sections will go into detail about the trade-off between extra data to be stored versus gain in approximation rate. Section 3.4.1 will briefly consider issues concerning a fixed splitting rule. Section 3.4.2 will be devoted on using an adaptive splitting approach.

3.4.1 Fixed topology

When the splitting rule is fixed in advance, no extra topological information has to be stored, since the connection of new piercing points always happens in the same way. On the other hand the triangle splitting is blind with respect to the shape of the above lying surface connected as defined by f. This already lets us assume that in the neighborhood of a contour, the discontinuity will not be taken into account. This means that edges are not discouraged to cross the contour. This is exactly what we wanted to circumvent as it were the main problem using tensor product wavelets, i.e. reducing the subdomains that come into contact with the contour.

3.4.2 Adaptive topology

At the other side, the way piercings points are connected with each other can be made data dependent , making the splitting rule *geometrically-adaptive*[3]. This means that the refinement operators produce different kinds of interconnections between $p_{j+1,k}$ depending on the geometry of f. The downside is that we have to store some topological information, since the search for new piercing points depend on the interconnections used. Adaptive triangular subdivision can be favourable, as long as this extra amount of storage pays off in terms of the decay of n-terms approximation error ϵ_n. During decomposition, each time a triangle has to be refined, a selection of how to connect the piercing points p_{j+1} has to be made. Such a routine demands extra computational effort to select for the *best* subdivision, *'best'* according to some *penalty functions* $\Xi(f, \mathcal{M}_j(E_j))$ i.e.:

$$E_j^* = \arg\min_{E_j} \Xi(f, \mathcal{M}_j(E_j)),$$

with $\mathcal{M}_j(E_j)$ the abbreviate for $\mathcal{M}_j(V_j, E_j, F_j)$, the mesh defined by the piercing points V_j, edges E_j and the faces F_j induced by E_j and V_j. Also this leads to a trade-off between time complexity (computing speed) versus rate of approximation (compression). For Ξ we could chose between different kinds of penalty functions, e.g. using an L_p norm

$$\Xi = \|f - \mathcal{M}_j(E_j)\|_{L_p}, \tag{8}$$

[3]Also the search for a piercing point introduces a nonlinearity depending on f (Equation (5))

using an Hausdorff distance H_{L_p}, the right hand side of Eq. (8) becomes $H_{L_p}(f, \mathcal{M}_j(E_j))$ or using the oriented (or single-sided) Hausdorff distance h_{L_p}, i.e. $h_{L_p}(f, \mathcal{M}_j(E_j))$ or using a (discrete) curvature measure.

Note. *Considering the nature of the normal offset method, namely measuring the distance in the normal direction between certain points of the approximating mesh and f, it would be more sound to use a single sided Hausdorff distance. The Hausdorff distance method, however, is computationally expensive to calculate. We shall use the L_p norm in this paper for simplicity.*

In this paper we propose an adaptive splitting algorithm that can chose the best *conformal* quadtree splitting as depicted in Figure 3. From Figure 4 we can already sense the need for an adaptive triangular subdivision depending on f, to obtain the best approximation rate, as will be studied in Section 5.2.

4 Modifications towards digital images

We define a digital image as a function $\overline{f} : \mathbb{N}^2 \mapsto \mathbb{Z}$ which maps a finite set of elements $\overline{\Omega} := \{(x_i, y_i)\} \in \mathbb{N}^2$ to a discrete set of function values $\in \mathbb{Z}$. Working with a discrete set of function values introduces some peculiarities. The remainder of this section will be devoted towards problems due to working with both a discrete set of pixels making up the domain and the function values constrained to lie in $\mathbb{Z}$.

Problem 1 (discrete prediction operator)**.** Uptil now, we worked on a continuous domain $\Omega \subset \mathbb{R}^2$. This allowed us to define the prediction operator $\mathcal{P}$ by Equation (2). If the function $\overline{f}$ is only defined on a finite subset of $\mathbb{N}^2$,however, the prediction operator $\mathcal{P}$ should map $\mathbf{x} \in \mathbb{N}^2$ to $\mathbf{x}^* \in \mathbb{Q}^2$. As a consequence, $\Delta x_{j,k} \notin 2\mathbb{N}$ or $\Delta y_{j,k} \notin 2\mathbb{N}$ implies $\mathbf{x}^*_{j+1,2k+1} \notin \mathbb{N}^2$. This becomes a problem when discretizing the edges (Problem 2).

Solution 1. We could round the predicted location $\mathbf{x}^*_{j,k}$ predicted by $\mathcal{P}$, such that $\overline{\mathbf{x}}^*_{j,k} = \text{round}\left(\mathbf{x}^*_{j,k}\right) \in \mathbb{N}^2$. However, this causes the partition not to be disjoint any longer, which results in crosscutting edges of the triangular subdomains as depicted in Figure 5. For a proper discretization of the predictor operation we first have to come up with a proper edge discretization.

Definition 4.1. *We define a discrete edge $\overline{\mathbf{e}}_{\mathbf{x}}(p_{j,k}, p_{j,k+1})$ as the set of pixels $\mathbf{i} \in \mathbb{N}^2$* joining *$\mathbf{x}_{j,k}$ with $\mathbf{x}_{j,k+1}$, ordered with respect to increasing distance from the starting point $\mathbf{x}_{j,k}$. Thus $\overline{\mathbf{e}}_{\mathbf{x}}(p_{j,k}, p_{j,k+1}) = \left\{\mathbf{i}(\ell) \in \mathbb{N}^2 \ : \ \ell = 0, \ldots, L-1\right\}$ with $\mathbf{i}(0) = \mathbf{x}_{j,k}$ and $\mathbf{i}(L-1) = \mathbf{x}_{j,k}$, while dist$(\mathbf{x}_{j,k}, \mathbf{i}(\ell)) <$ dist$(\mathbf{x}_{j,k}, \mathbf{i}(\ell+1))$, $\ell = 0 \ldots L-2$.* Joining *means that $\mathbf{i}(\ell) = \mathbb{B}(\ell, \mathbf{x}_{j,k}, \mathbf{x}_{j,k+1})$ with $\mathbb{B}(.,.,.) : (\mathbb{N}, \mathbb{N}^2, \mathbb{N}^2) \mapsto \mathbb{N}^2$.*

For $\mathbb{B}$ we choose the Bresenham algorithm [1] which is a fast and popular line rasterisation algorithm extensively used computer graphics. We replace the prediction operator $\mathcal{P}$ by the discretized prediction operator $\overline{\mathcal{P}}$ defined as:

$$\overline{\mathcal{P}}(\mathbf{x}_{j,k}, \mathbf{x}_{j,k+1}) = \mathbf{i}(\ell^*) = \mathbf{x}^*_{j+1,2k+1} \ , l^* = \left\lfloor \frac{L}{2} \right\rfloor, \ L := |\{\mathbf{i}(\ell)\}| \ , \mathbf{i}(\ell) = \mathbb{B}(\ell, \mathbf{x}_{j,k}, \mathbf{x}_{j,k+1}), \tag{9}$$

with $|X|$ denoting the cardinality of the set X. This means that x^* is selected as the couple $\mathbf{i} \in \mathbb{N}^2$ that is in the middle of the ordered set $\{\mathbf{i}(l)\}$. Next, we have to discretise the constraint (4), and substitute it with a discrete constraint i.e. :

$$\mathbf{x}_{j+1,2k+1} \in \{\mathbf{i}(\ell)\}, \ \ \mathbf{i}(\ell) = \mathbb{B}(\ell, \mathbf{x}_{j,k}, \mathbf{x}_{j,k+1}) \tag{10}$$

This requires to *search* along the set of pixels defined by Equation (10), to find the $\tilde{\ell}$ that satisfies Equation (3).

Problem 2 (discrete edge refinement)**.** In the continuous setting, consider two piercing points $p_{j,k}$ and $p_{j,k+1}$ $\in \mathbb{R}^3$ defining an edge $\mathbf{e}(p_{j,k}, p_{j+1,k})$ as

$$\mathbf{e}(p_{j,k}, p_{j,k+1})(t) = t(p_{j,k}) + (1-t)(p_{j,k+1}), t \in [0,1],$$

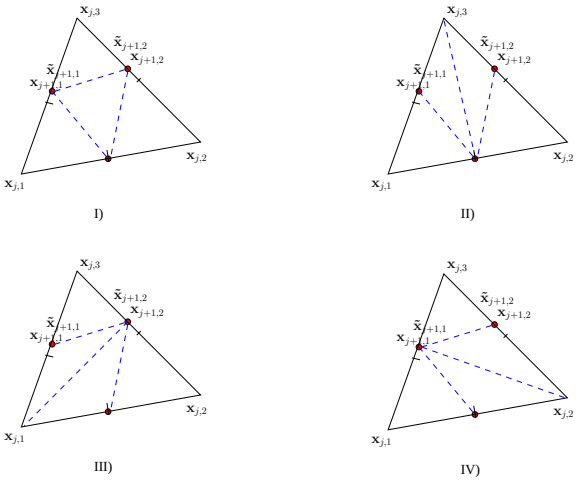


Figure 3: *The adaptive splitting algorithm is constrained to choose among one of these four splittings*

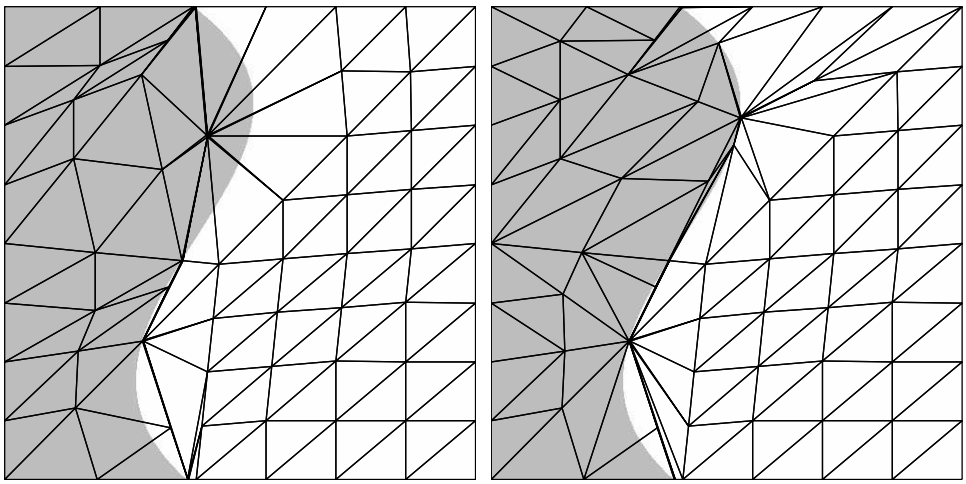


Figure 4: *Both figures started out with a base mesh consisting of two triangles (i.e. ⧅), with the common edge the diagonal starting from the upper left corner to the lower right. (Left) The pictures shows the resulting triangulation of a fixed splitting method at resolution level $j = 3$. (Right) Triangulation resulting from an adaptive splitting method (using penalty function of Eq. 8) at resolution level $j = 3$. As we can see, there are few triangle edges crossing the contour with the adaptive splitting method in comparison with a fixed one.*

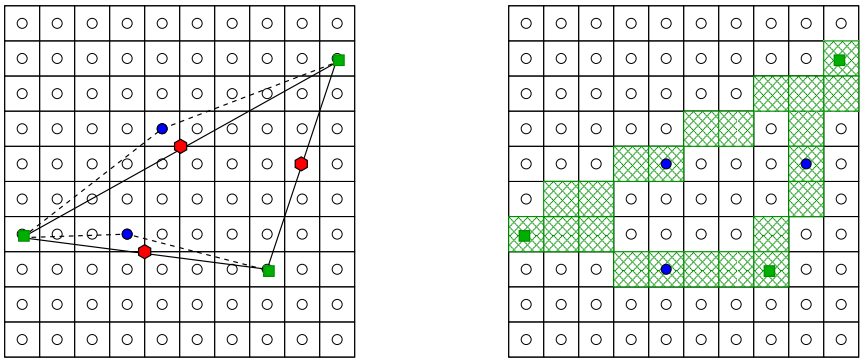


Figure 5: *(Left) Due to roundoff effects, pixels that first lied outside a triangle can get absorbed by it, while other pixels get lost by the same triangle after a refinement step. This, in its turn, may cause a non disjunct set of triangles with overlapping edges or even pixels no longer belonging to any triangle (i.e. gaps). The hexagonal shaped icons are the ones predicted with the continuous prediction operator $\mathcal{P}$, while the filled circles are the rounded values of the hexagonal ones towards the closest pixel location. (Right) The proposed solution by first discretizing the edge and taking the middle pixel of the discrete path between the outer edge pixels. This way, the predicted location will always belong to the discrete edge, obtaining a partitioning without gaps or overlapping areas (only the edges of adjacent triangles will have all their pixels in common).*

and the projection $\mathbf{e}_{\mathbf{x}}(p_{j,k}, p_{j+1,k})$ of $\mathbf{e}(p_{j,k}, p_{j+1,k})$ on the domain as follows:

$$\mathbf{e}_{\mathbf{x}}(p_{j,k}, p_{j,k+1})(t) = t(\mathbf{x}_{j,k}) + (1-t)(\mathbf{x}_{j,k+1}), t \in [0,1],$$

$\mathbf{x}_{j,k}$ and $\mathbf{x}_{j,k+1} \in \mathbb{R}^2$. Equation (5) forces the piercing point $p_{j+1,2k+1}$ to lie on $\mathbf{e}_{\mathbf{x}}(p_{j,k}, p_{j,k+1})$. Constructing subedges $e(p_{j,k}, p_{j+1,2k+1})$, $e(p_{j+1,k}, p_{j,k+1})$ and applying this strategy recursively on these edges, all piercing points will still be found on $\mathbf{e}_{\mathbf{x}}(p_{j,k}, p_{j,k+1})$. This is what we meant by Equation (5) being an *edge refining method*.

Solution 1 does not preserve this property in the discrete setting. Since it is not necessarily the case that

$$\overline{\mathbf{e}}_{\mathbf{x}}(p_{j,k}, p_{j+1,2k+1}) \subset \overline{\mathbf{e}}_{\mathbf{x}}(p_{j,k}, p_{j,k+1}) \text{ and } \overline{\mathbf{e}}_{\mathbf{x}}(p_{j+1,2k+1}, p_{j,k+1}) \subset \overline{\mathbf{e}}_{\mathbf{x}}(p_{j,k}, p_{j,k+1}), \tag{11}$$

in which $\overline{\mathbf{e}}_{\mathbf{x}}(p_{j,k}, p_{j+1,2k+1})$ and $\overline{\mathbf{e}}_{\mathbf{x}}(p_{j+1,2k+1}, p_{j,k+1})$ are also constructed by the Bresenham algorithm.

Solution 2. In what follows let $p_{j+1,2k+1} = \overline{\mathcal{P}}(p_{j,k}, p_{j,k+1})$. In order to provide a solution for Problem 2 satisfying Equation (11) we proceed as follows:

if $\neg\,(\exists E') \in E_j \;\mid\; (p_{j,k}, p_{j,k+1}) \in E'$ **then**
$\bar{\mathbf{e}}_{\mathbf{x}} = \{\mathbb{B}(\ell, \mathbf{x}_{j,k}, \mathbf{x}_{j,k+1})\}$
else
we *allocate* a subset of pixels defined by $\overline{\mathbf{e}}_{\mathbf{x}}(p_{j,k}, p_{j,k+1})$ to its child edges by:

$$\overline{\mathbf{e}}_{\mathbf{x}}(p_{j,k}, p_{j+1,2k+1}) \leftarrow \left\{\mathbf{i}(\ell) \;:\; \ell = 0 \cdots \tilde{\ell},\;\; \mathbf{i}(\ell) = \mathbb{B}(\ell, \mathbf{x}_{j,k}, \mathbf{x}_{j,k+1})\right\} \tag{12}$$

$$\overline{\mathbf{e}}_{\mathbf{x}}(p_{j+1,2k+1}, p_{j,k}) \leftarrow \left\{\mathbf{i}(\ell) \;:\; \ell = \tilde{\ell} \cdots L-1,\;\; \mathbf{i}(\ell) = \mathbb{B}(\ell, \mathbf{x}_{j,k}, \mathbf{x}_{j,k+1})\right\} \tag{13}$$

end if

Problem 3 (pixel selection)**.** Even if we start from a location $x^* \in \mathbb{N}^2$ a discrete piercing point satisfying the nonlinear Equations (3) and (4) will almost never be found, since at every discrete point $\in \mathbb{N}^2$ the function only takes on discrete values $\in \mathbb{Z}$, i.e. $\overline{f} : \mathbb{N}^2 \mapsto \mathbb{Z}$. So even if the function $\overline{f}$ behaves rather smooth, the normal ray will still make an error when predicting the exact function value as is will take on a value in $\mathbb{R}$. In particular Equation (3) is hard to satisfy.

Solution 3. Under the assumption that $z_{j,k} < z_{j,k+1}$,we replace condition (3), by looking for an $\tilde{\ell}$ such that:

$$\begin{array}{rcll} \text{sign}\left(\overline{f}\left(\mathbf{i}(\hat{\ell})\right) - n\left(\hat{\ell}\right)\right) & = & -\text{sign}\left(\overline{f}\left(\mathbf{i}(\tilde{\ell}+1)\right) - n\left(\hat{\ell}+1\right)\right) & \text{if } 0 \leq \hat{\ell} \leq \ell^* \\ & \text{or} & & \\ \text{sign}\left(\overline{f}\left(\mathbf{i}(\hat{\ell}-1)\right) - n\left(\hat{\ell}-1\right)\right) & = & -\text{sign}\left(\overline{f}\left(\mathbf{i}(\hat{\ell})\right) - n\left(\hat{\ell}\right)\right) & \text{if } \ell^* < \hat{\ell} \leq L-1 \end{array} \tag{14}$$

and define $\tilde{\ell} := \arg\min_{\hat{\ell}} \left\{ |\hat{\ell} - \ell^*| \right\}$

with

$$n(\ell) := \overline{f}\left(\mathbf{i}\left(\ell^*\right)\right) + round\left(\frac{f_{j,k} - f_{j,k+1}}{\|\mathbf{x}_{j,k} - \mathbf{x}_{j,k+1}\|} \left\| \mathbf{i}(\ell) - \mathbf{i}\left(\tilde{\ell}\right) \right\| \right)$$

and

$$\mathbf{i}(\ell) = \mathbb{B}(\ell, \mathbf{x}_{j,k}, \mathbf{x}_{j,k+1}).$$

The distinction between cases $0 \leq \hat{\ell} \leq \ell^*$ and $\ell^* < \hat{\ell} \leq L-1$ will play an important role in the error decay in the digital setting as will be made clear in Subsection 5.2.2.

Because $\overline{f}\left(\mathbf{x}_{\mathbf{i}(\tilde{\ell})}\right) - n\left(\tilde{\ell}\right)$ will generally differ from zero, we have to bring in an extra coefficient.

Definition 4.2 (vertical offset)**.** *Let $n, \mathbf{i}(.), \overline{f}$, as previously defined. We define a vertical offset $v_{j+1,2k+1}$ as:*

$$v_{j+1,2k+1} = \overline{f}\left(\mathbf{i}(\tilde{\ell})\right) - n\left(\tilde{\ell}\right) \tag{15}$$

This *vertical offsets* will be stored together with the *normal offset* for later reconstruction.

As the normal offset together with the vertical offset determine the real function value, it is important in the two dimensional setting to make the distinction when the piercing point is found on the increasing side of the normal ray, or on the decreasing side beginning at the predicted point as expressed in Eq. (14) and Eq. (15). This will be of great importance when normal rays pierce through discontinuities in the two dimensional setting. The equations imply that, from whatever side the discontinuity is pierced, on images of the horizon class $\mathcal{H}$, the normal offset combined with the vertical offsets always results in the same function value (See Figure 6).

Remark 4.1. *In the light of Problem 4 the* normal offset *will be used to indicate the position $\mathbf{x}_{j+1,2k+1}$ of the piercing point $p_{j+1,2k+1}$ and an approximate of $\overline{f}(\mathbf{x}_{j+1,2k+1})$, using nothing but one integer value. While the* vertical offset *$v_{j+1,2k+1}$ will be used to correct the function value, or even slightly adapted to $\tilde{v}_{j+1,2k+1} \in \mathbb{R}$ in order reduce the overall n-terms approximation error ϵ_n.*

In general the extra vertical offsets needed to exactly represent the function values are large only when pierced through a discontinuity. When the normal rays pierce through smooth areas, the piercing points function value $z_{j,k}$ will also be close to the function value $\overline{f}(\mathbf{x}_{j,k})$. In this way the vertical offsets are small and sparse where the function behaves smoothly. In parts where the functions have a more versatile behavior more of those offsets are likely to have a significant value.

Problem 4 (iteration)**.** In the continuous setting only a normal offset (Equation (**??**)) sufficed to exactly pinpoint the piercing point $p_{j+1,2k+1}$ located on f, based on previous piercing points. As for the discrete setting we learned from Problem 3 that at least two *offsets* are needed to locate a point on $\overline{f}$. The question remains whether to set

a) $p_{j+1,2k+1} = n(\tilde{l}) + \mathbf{e}_z v_{j+1,2k+1}$

or

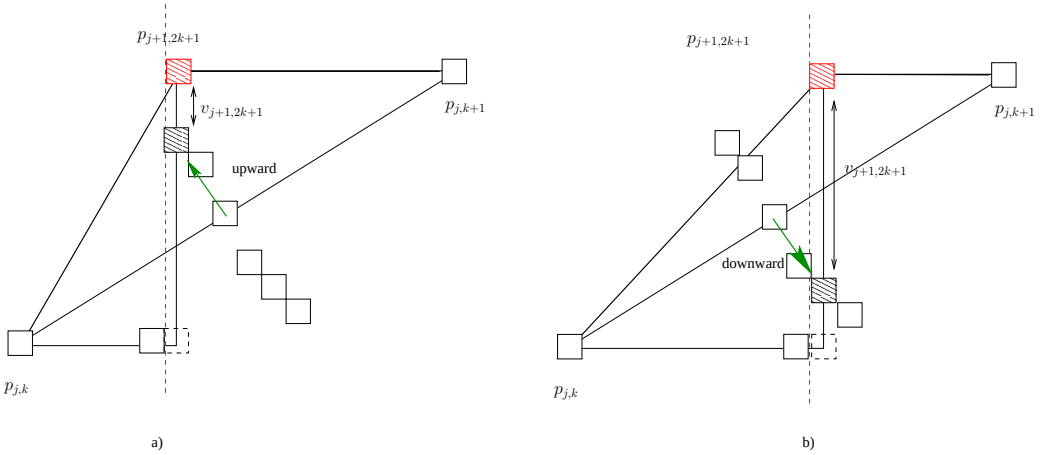


Figure 6: *(left) Moving upward: stop right* before *discontinuity. (Right) Moving downward: stop right* after *discontinuity.*

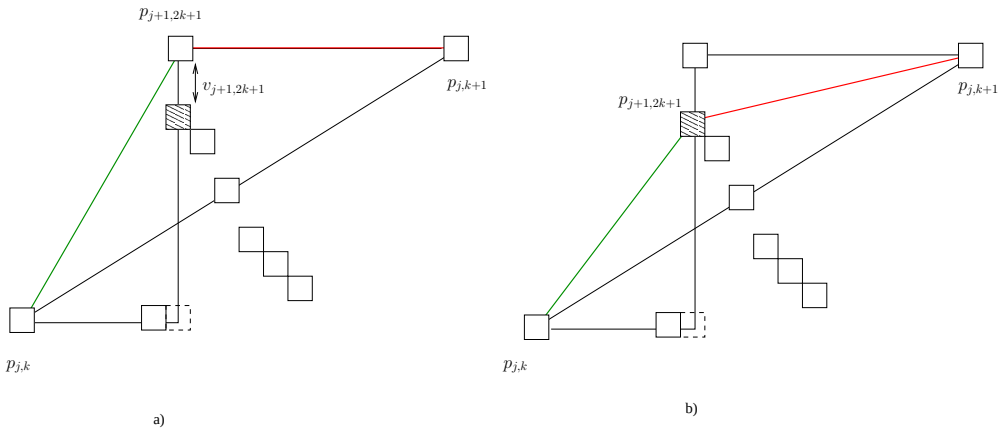


Figure 7: *As we can see, in case of the Horizon class, option a) gives already the exact solution for the right part of the function. But as we will like to construct an algorithm to handle more general functions, option b) is equally likely to perform as well as a) , because we do not know anything about the function except the location where it is being pierced by the normal ray.*

b) $p_{j+1,2k+1} = n(\tilde{l})$

to build $\mathcal{M}_{j+1}$, whose edges serve to define *normal rays* in a next iteration. Those two ways are depicted in Figure 7.

Solution 4. For stability considerations as well as for optimization aspects, option b) seems the most suited. The mesh $\mathcal{M}_j$ will be constructed using only the normal offsets. This leads to a separation of *normal* and *vertical offsets*. This separation makes the transform more *stable* with respect to pertubations on the vertical offsets. It even allows to alter the vertical offsets without touching the partition imposed by the normal offsets in order to *optimize* the reconstructed image from a selected subset of normal and vertical offsets.

5 Compression

In the context of data reduction there are two kinds of compression. On the one hand, *lossless* compression schemes reduce the amount of data needed to reconstruct the original one without any loss of information. This lossless data reduction has a limit, in the sense that no compression method exists that can represent the original data with less than a specific number of bits b^*. This bound is well known from the field of information theory. If a certain loss of information is acceptable *lossy* compression schemes try to encode the original data with a certain compression rate in a way that the 'least' important information is thrown away first. Which information to be preserved depends on the problem at hand.

5.1 Lossless compression

To losslessly reduce the amount of data, *all* coefficients should be stored with the smallest possible number of bits. The theoretical lower bound of the bitrate r, i.e. the required number of bits per symbol, can be approximated when taking the probability of the occurrence of each coefficient value into account. This probability density function (PDF) must be known in advance. Entropy encoders , like Huffman encoders, map each value of a *finite* alphabet $\mathbb{A}$ to a string of bits, called codewords, i.e. $C : \mathbb{A} \to \{0,1\}^r$ such that the bitrate is minimized.

It was pointed out in Solution 3 (Section 4) that the signed length of the normal offset $\|\vec{n}_{j,k}\|$, which takes values in $\mathbb{R}$, could be replaced by an integer index $\tilde{l} \in \mathbb{Z}$ to reconstruct $p_{j+1,2k+1}$ from $p^*_{j+1,2k+1}$. As turns out later, the set of values from which $\tilde{l}$ is drawn is bounded and those bounds are known *a priori*. From the solution for Problem 4, it is clear that an extra vertical offset is needed to obtain the real function value $f(\mathbf{x}_{j+1,2k+1})$. Since the vertical offsets might be adjusted in a postprocessing step taking on values of $\mathbb{R}$, they do not necessarily come from a finite set and are thereby not suited for lossless compression.

Note. *The discretization process could also be accomplished when constructing a discrete line in the three dimensional case and expressing the piercing points by the number of voxels*[4] $\iota_{j+1,2k+1}$ *to be traversed along the rasterized line starting from the* $p^*_{j+1,2k+1}$ *of the edge till the piercing point* $p_{j+1,2k+1}$ *is reached. The sign denotes in which direction to travel. The main disadvantage of this method is that the discretizes 3D line cannot be seen as a function any longer, since it is not restricted to one function value corresponding to its pixel.*

For the sake of notational simplicity we introduce the index $\iota_{j+1,2k+1}$ as:

$$\iota_{j+1,2k+1} := \ell^* - \left\lceil \frac{L}{2} \right\rceil$$

with ℓ^*, L as defined in Solution 1 and 3 (Section 4). Define the set of possible values of ι as:

$$\mathbb{A} := \{-\lceil L/2 \rceil, \ldots, \lceil L/2 \rceil - 1\} \subset \mathbb{Z}$$

[4] a voxel is the 3D equivalent of a pixel

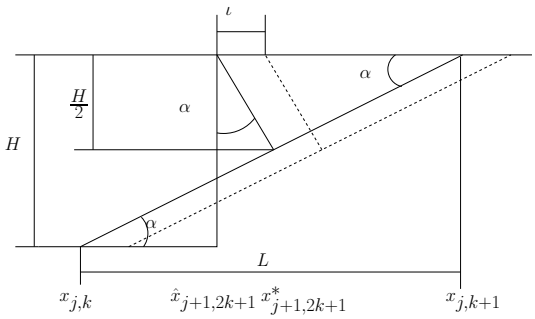


Figure 8: *Crosscut of an image* $\in \mathcal{H}$ *containing a contour.*

It can easily be seen that $\iota_{j+1,2k+1}$ is bounded in terms of L, 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 decrease. These normal indices $\iota_{j+1,2k+1}$ can be seen as *random* variables $\mathbf{X}$ with values coming from an alphabet $\mathbb{A}$ of size L.
Assume that we can model this source of information as instances of a random distribution. Then, every symbol from $\mathbb{A}$ has a probability p_i. The amount of information contained in a symbol sequence drawn from $\mathbb{A}$) is measured by the entropy $\mathcal{S}$:

$$\mathcal{S} = -\sum_{i=0}^{L-1} p_i \log_2(p_i) \tag{16}$$

For a comprehensive reading about information theory we refer to the paper of C.E. Shannon [12]. 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 L possibilities with $\log_2 L$ bits and an equal probability, $\mathcal{S}$ will reach its maximum value $\mathcal{S}_{\max} = \log_2 L$.

We now construct an ***a priori***[5] PDF (Probability Density Function) of these normal indices when given two edge points $p_{j,k}$ and $p_{j,k+1}$, for the Horizon class of images. Figure 8 will be used in Proposition 5.2 as well as in the rest of the paper.

Proposition 5.1. *If we define* $\hat{\mathbf{x}}_{j+1,2k+1}$ *as the point on* $e_{\mathbf{x}}(p_{j,k}, p_{j,k+1})$ *where the discontinuïty is situated then:*

$$\|\mathbf{x}_{j+1,2k+1} - \hat{\mathbf{x}}_{j+1,2k+1}\| \leq \left\|\mathbf{x}^*_{j+1,2k+1} - \hat{\mathbf{x}}_{j+1,2k+1}\right\|$$

The proof of Proposition 5.1 follows easily from Figure 8.

Proposition 5.2. *Consider* $f_\Omega(x, y) \in \mathcal{H}$*, where* $\Omega = \square \subset \mathbb{R}^2$*, and two points* $\mathbf{x}_{j,k}, \mathbf{x}_{j,k+1} \in \Omega$

a) *Assume* $f(\mathbf{x}_{j,k}) = 0$ *and* $f(\mathbf{x}_{j,k+1}) = H$.

b) *Let* d *be defined as the distance between* $\mathbf{x}_{j,k}$ *and the discontinuity, as depicted in Figure 8.*

c) *Let* d *be uniformly distributed on the interval* $[0, L]$ *to be consistent with a).*

d) *Assume the distance* $L := \|\mathbf{x}_{j,k+1} - \mathbf{x}_{j,k}\|$ *and* H *both* fixed, *with* $H < L$

[5] *a priori* since at the decompression stage we only have coarse level data at hand.

Then the PDF of $\iota_{j+1,2k+1} \in \mathbb{R}$ will be:

$$p(\iota_{j+1,2k+1}) = \begin{cases} \dfrac{1}{L}, & -\dfrac{H\tan\alpha}{2} < \iota_{j+1,2k+1} < \dfrac{H\tan\alpha}{2} \\ \dfrac{1}{2}\left(1 - \dfrac{H\tan\alpha}{L}\right), & \iota_{j+1,2k+1} = \pm\dfrac{H\tan\alpha}{2} \\ 0, & |\iota_{j+1,2k+1}| > \dfrac{H\tan\alpha}{2} \end{cases}$$

Proof. The proof follows readily from Figure 8 □

Corollary 5.1. *Making an analogue assumtion as in Proposition 5.2 for the discrete setting i.e. $f_\Omega(x,y) \in \mathcal{H}$, where $\Omega = \square \subset \mathbb{R}^2$, and two points $\mathbf{x}_{j,k}, \mathbf{x}_{j,k+1} \in \mathbb{N}^2 \subset \Omega$, $L \notin 2\mathbb{Z}$*

a) *Assume $f(\mathbf{x}_{j,k}) = 0$ and $f(\mathbf{x}_{j,k+1}) = H$.*

b) *Let d be defined as the number of pixels to be traversed between $\mathbf{x}_{j,k}$ and the discontinuity, as depicted in Figure 8.*

c) *Let d be uniformly distributed on the set $\{0 \dots L\}$ to be consistent with a).*

d) *Assume the L and H as defined in Equation (9), both* fixed, *with $H < L$*

Then the PMF (Probability Mass Function) of $\iota_{j+1,2k+1} \in \mathbb{Z}$ will be:

$$p(\iota_{j+1,2k+1}) = \begin{cases} \dfrac{1}{L}, & \iota_{j+1,2k+1} \in \left\{-\left\lfloor\dfrac{H\tan\alpha}{2}\right\rfloor, \dots, \left\lfloor\dfrac{H\tan\alpha}{2}\right\rfloor\right\} \\ \dfrac{1}{2}\left(1 - \dfrac{1}{L}\left(2\left\lfloor\dfrac{H\tan\alpha}{2}\right\rfloor + 1\right)\right), & \iota_{j+1,2k+1} = \pm\left\lceil\dfrac{H\tan\alpha}{2}\right\rceil \\ 0, & |\iota_{j+1,2k+1}| \geq \left\lfloor\dfrac{H\tan\alpha}{2}\right\rfloor \end{cases}$$

Corollary 5.2. *Under the same conditions as Corollary 5.1, for the Horizon class of functions $\mathbb{A}^*$ is a* finite *set of symbols, i.e.*

$$\mathbb{A} := \left\{-\left\lceil\frac{H\tan\alpha}{2}\right\rceil, \dots, \left\lceil\frac{H\tan\alpha}{2}\right\rceil\right\},$$

with $|\mathbb{A}| \leq L$.

Proof. This immediately follows from Corrolary 5.1. □

Proposition 5.3. *For a fixed ratio H/L the gain $\mathcal{G}_L(H/L) := \dfrac{\mathcal{S}_L^{max}(H/L)}{\mathcal{S}_L(H/L)}$, decreases monotonically towards $\dfrac{L^2}{H^2}$ as the length L (in number of pixels) tends to ∞:*

$$\lim_{L\to\infty} \mathcal{G}_L(H/L) = \lim_{L\to\infty} \frac{\mathcal{S}_L^{max}(H/L)}{\mathcal{S}_L(H/L)} = \frac{H^2}{L^2}$$

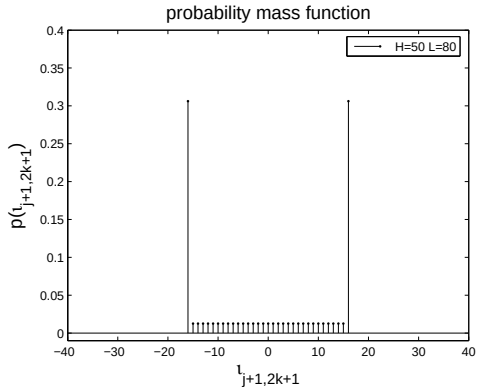


Figure 9: *The picture shows a PMF of $\iota_{j+1,2k+1}$ of a function $\in \mathcal{H}$ where $H = 50$ and L equals 60*

Proof. In the continuous setting Equation (16) becomes:

$$\mathcal{S} = -\int_{t=-L/2}^{L/2} p(t)\log_2(p(t)) = -\frac{\ln\left(L^{-1}\right)H^2}{L^2\ln(2)} + 1 \tag{17}$$

from which we see that, if we fix H/L the entropy shall rise when L does. With this in mind and Corollary 5.3, we clearly see that the estimated bit rate grows, as the resolution of our digital images rises, if we fix H/L, only considering edges of length L. However, the gain converges towards:

$$\mathcal{G}_\infty(H/L) = \lim_{L\to\infty}\left(-\frac{\ln\left(L^{-1}\right)H^2 + L^2\ln(2)}{L^2\ln(L)}\right)^{-1} = \lim_{L\to\infty}\left(\frac{H^2}{L^2} - \frac{1}{log_2(L)}\right)^{-1} = \frac{L^2}{H^2}$$

as can be seen from the Figure 10 in the right picture. □

Corollary 5.3. *With the same notation of Proposition 5.3, set*

$$H'/L' = H''/L'',$$

then

$$L' < L'' \Rightarrow \mathcal{S}_L(H'/L') < \mathcal{S}_L(H''/L'')$$

The entropy encoder generates a reduction of data without loss of information. Even without distortion we can lower the bitrate (bits per pixel). This lossless compression is a full gain to the bit rate, as it reduces the bitrate at the very beginning of a rate-distortion curve where no distortion is yet introduced. Figure 11 schematically represents 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.

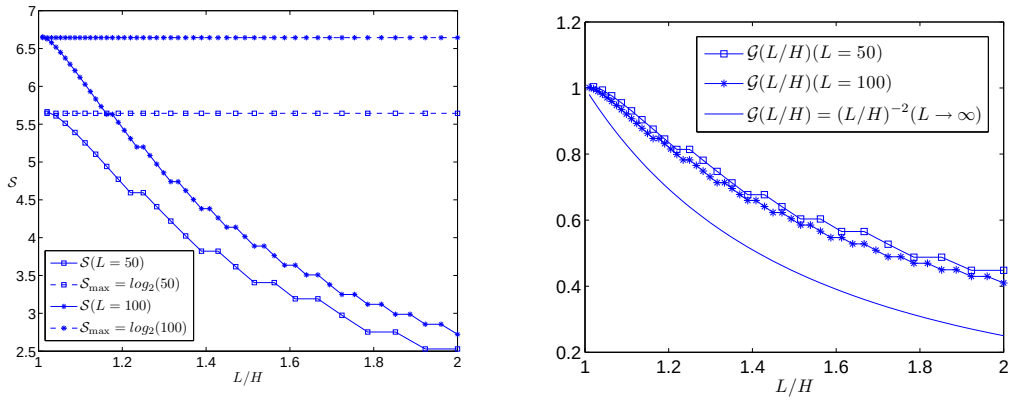


Figure 10: *(Left) The picture shows the plot for the entropy $\mathcal{S}$ as defined by Equation 16 for two different values of L (i.e. $L = 50, L = 100$) and H within the range of $[L/2, L-1]$. This is equal to a change of resolution when looking at the values of $\mathcal{S}$ at the same ordinate. Mark that the entropy according to the greatest $L = 100$ is always larger than for $L = 50$ at the same L/H value. (Right) The picture shows the $\mathcal{G}(L/H)$ for the values of L in the left picture. The picture shows the monotonic convergence of $\mathcal{G}$ towards the limit function as L gets larger.*

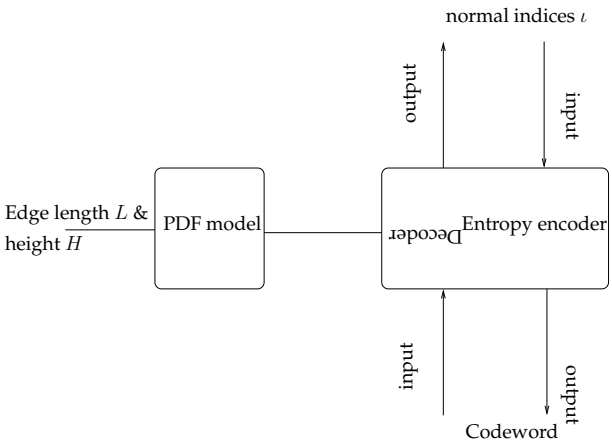


Figure 11: *Schematic illustration of the entropy encoder and decoder. On the basis of the data of the edge, the normal indices can be mapped to codewords and visa versa.*

5.2 Lossy compression

Error decay of the proposed method for images $\in \mathcal{H}$

This section examines how fast the refinement procedure finds piercing points on the contour, starting from the edges e_1 and e_2 as depicted on the left picture of Figure 12. Furthermore, we establish a refinement procedure that acts as an adaptive triangulation on each triangle such that the newly formed edges align themselves parallel along the contour. We assume that the contour $c(x, y)$ is smooth enough to approximate it within each triangle by $c(x, y) \simeq c(x_0, y_0) + \frac{\partial c}{\partial x}_{|x_0}(x - x_0) + \frac{\partial c}{\partial y}_{|y_0}(y - y_0)$. The following theorem examines the probability that the refinement results in 0 (Subcase III), 1 (Subcase II), or 2 (Subcase I) piercing points found on the contour, depending on the ratios $\frac{H}{L_{j,i}}$. We refer to Figure 12 to have a visual on the situation.

Corollary 5.4. *Let L, H, and d be defined as in Proposition 5.2, $\eta_{j,i} := \dfrac{H}{L_{j,i}}$. The probability that 0,1 or 2 ordinates $\mathbf{x}_{j,1}, \mathbf{x}_{j,2}$, according to piecing points $p_{j,1}, p_{j,2}$, will satisfy $c(\mathbf{x_{j,i}}) \equiv 0,\ i \in \{1, 2\}$, under the assumption that distances d_1 and d_2 are uniformly distributed on $[0, L_{j,1}]$ and $[0, L_{j,2}]$, equals:*

	Case A $\eta_{j,1} \leq 1, \eta_{j,2} \leq 1$	*Case B* $\eta_{j,1} \geq 1, \eta_{j,2} \geq 1$	*Case C* $\eta_{j,2} \leq 1 \leq \eta_{j,1}$
$c(\mathbf{x_{j,1}}) = 0 \& c(\mathbf{x_{j,2}}) = 0$	$(\eta_{j,1}\eta_{j,2})^2$	1	$\eta_{j,2}^2$
$c(\mathbf{x_{j,1}}) = 0 \oplus c(\mathbf{x_{j,2}}) = 0$	$\eta_{j,1}^2 + \eta_{j,2}^2 - (\eta_{j,1}\eta_{j,2})^2$	0	$1 - \eta_{j,2}^2$
$c(\mathbf{x_{j,1}}) \neq 0 \& c(\mathbf{x_{j,2}}) \neq 0$	$1 - (\eta_{j,1}^2 + \eta_{j,2}^2)$	0	0

Proof. *a*) We prove situation A.*I*), which follows almost immediately from Proposition 5.2. The others are analogue.

$$\begin{aligned} p\left(c(\mathbf{x_{j,1}}) = 0\right) \ \& \ p\left(c(\mathbf{x_{j,2}}) = 0\right) &= p\left(d_{j,1} \in \left[\frac{L_{j,1}}{2} \pm \frac{H^2}{2L_{j,1}}\right] \ \& \ d_{j,2} \in \left[\frac{L_{j,2}}{2} \pm \frac{H^2}{2L_{j,2}}\right]\right) \\ &= \frac{H^2}{L_{j,1}^2}\frac{H^2}{L_{j,2}^2} \\ &= (\eta_{j,1}\eta_{j,2})^2 \end{aligned}$$

b) From the construction in Figure 8 the piercing point will always be found on the discontinuïty.

c) The result follows from a) and b).

□

In the next two subsections we have a closer look at Subcases III) and I). Subcase II) can be derived analogously.

5.2.1 Subcase III): $c(\mathbf{x_{j,1}}) \neq 0 \ \& \ c(\mathbf{x_{j,1}}) \neq 0$

We first concentrate on one side of the triangle crossed by the contour as depicted on the left of Figure 12. Following propositions concentrate on a 1-D analysis. In this paragraph we assume that the following is always satisfied:

$$H \leq \min\{L_1, L_2\}. \tag{18}$$

We only perform an elaborated study of situation A.III) since this is the only situation that also applies to the discrete setting (B.III) and C.III) have zero probability). We investigate how fast the piercing points are attracted towards the line discontinuity in that part of the edge where both end points have different z-values. The following Proposition investigates a worst case scenario in the sense that the contour is located with respect

to the initial vertices such that the normal ray has never pierced through the line discontinuity for $\tilde{j} < j$. This explains why H in Eq. 18, remains constant in the rest of this section. Once $\eta_j \geq 1$, the location of the next piercing point located on the contour is a certitude.

Proposition 5.4. *In the knowledge that $H/L_j < 1$ and for the same edge refinement the normal ray never pierces through the discontinuity for $\tilde{j} < j$, i.e., the indices ι_i in all previous refinements where all of length H^2/L_j, with L_j the length of the subedge which made an error, then*

$$\eta_{j+1} = 2\eta_j \frac{1}{\left(1-\eta_j^2\right)} \qquad \textit{with } \eta_j = \frac{H}{L_j}. \tag{19}$$

Proof. Obviously

$$\eta_{j+1} = \frac{H}{L_{j+1}} = \frac{H}{\frac{L_j}{2} - \frac{H}{2L_j}^2} = \frac{H}{\frac{L_j}{2}\left(1-\eta_j^2\right)} = 2\eta_j \frac{1}{\left(1-\eta_j^2\right)} \, .$$

□

Corollary 5.5. *From Proposition 5.4 it follows that:*

$$\eta_j < \frac{1}{2}\eta_{j+1} \, . \tag{20}$$

Moreover, we obtain an explicit formula forming a lowerbound for η_j:

$$\eta_j \geq \eta_0 2^j 2^{\left(C\left(4^j-1\right)\right)} \quad , \qquad C = \frac{\eta_0^2}{3\ln 2} \, . \tag{21}$$

Proof. From Equation (19) it follows that:

$$\begin{aligned}
\frac{\eta_j}{\eta_0} &= \prod_{i=0}^{j-1} 2\frac{1}{(1-\eta_i^2)} \\
&\geq 2^j \prod_{i=0}^{j-1} \frac{1}{\left(1-\left(2^i\eta_0\right)^2\right)} = 2^j \exp\left(-\sum_{i=0}^{j-1} \ln\left(1-\left(2^i\eta_0\right)^2\right)\right) && \text{(Eq. 20)} \\
&\geq 2^j \exp\left(\sum_{i=0}^{j-1} \left(2^i\eta_0\right)^2\right) = 2^j 2^{\left(C\left(4^j-1\right)\right)} \quad , \qquad C = \frac{\eta_0^2}{3\ln 2}
\end{aligned}$$

or

$$\frac{\eta_{j+1}}{\eta_j} \geq 2^{1+\tilde{C}4^j} \quad , \qquad \tilde{C} = \frac{\eta_0^2}{\ln 2}$$

from which also follows that

$$\frac{\eta_{j+1}}{\eta_j} \geq \frac{\eta_j}{\eta_{j-1}} \, .$$

□

From Corolarry 5.5 it is clear that this maximum number of refinements is obtained if the situation of Subcase III) remains in Case A as long as possible. The rise of η_j turns out to be super exponential, which shows it superiority to dyadical refinement methods to locate the discontinuity.

Corollary 5.6. *From the fact that $L_j/L_{j+1} = \eta_{j+1}/\eta_j$ it follows that*

$$\frac{L_j}{L_{j+1}} \quad \geq \quad 2^{1+\tilde{C}4^j} \qquad\qquad \tilde{C} = \frac{\eta_0^2}{\ln 2}$$

Proposition 5.4 and Corollary 5.5 express the strength of normal offsets in 1-d in finding point singularities. We now arrive at the main issue of the proposed method. The rest of this subsection concerns 2-d structures like triangle edges and line discontinuities, rather than piercing points and point singularities. The goal is to adaptively interconnect the piercing points in order to prevent newly formed edges to cross the contour. Following theorems can best be understood when referring to Figures 12 and 13.

Proposition 5.5. *For every possible cut of the contour through two triangle edges with respect to their piercing points (as depicted in Figure 12), there exists at least one edge $e' = e_{\mathbf{x}}(\mathbf{x}_\alpha, \mathbf{x}_\beta)$ $(\alpha, \beta \in \{j, j+1\} \times \{1, 2\})$ that must be produced by the splitting rule in order to minimize the number of triangles that come into contact with the contour in the following step. The edge that forms the subtriangle for the different situations is depicted in Figure 12 as the dashed line:*

a) $e' = e_{\mathbf{x}}(\mathbf{x}_{j+1,1}, \mathbf{x}_{j+1,2}) \Rightarrow 1$ *triangle is cut*

b) $e' = e_{\mathbf{x}}(\mathbf{x}_{j+1,1}, \mathbf{x}_{j+1,2}) \Rightarrow 3$ *triangles are cut*

c) $e' = e_{\mathbf{x}}(\mathbf{x}_{j+1,1}, \mathbf{x}_{j,2}) \Rightarrow 2$ *triangles are cut*

d) $e' = e_{\mathbf{x}}(\mathbf{x}_{j+1,2}, \mathbf{x}_{j,1}) \Rightarrow 2$ *triangles are cut*

Proof. As can be visually seen from Figure 12, the edge acts as a border separating the area for witch the function f will be perfectly approximated, and the area where a step function will be approximated by a plane face. □

Table 1 shows how many triangles would be cut by the contour if the interconnections where fixed, compared to the adaptive splitting approach (see Figure 13):

situation	fixed	adaptive
a)	1	1
b)	3	3
c)	3	**2**
d)	3	**2**

Table 1: *For each situation in Figure 13, the number of triangles being cut are given for a fixed splitting method, compared to an adaptive one.*

Especially situations c) and d) provide significant gain. Herein lies the strength of using an adaptive triangulation strategy.

From a theoretical point of view, we know (from the previous Proposition) which edge e needs to be in the partition. We try to split the triangle in such a way that in a next refinement the chance of the contour-crossed triangles having one or more piercing points lie on the discontinuity is large. From Corollary 5.4 we know that the chance a piercing point is found exactly on the discontinuity is proportional to $(H/L)^2$, where L is length of the side. Therefore, triangles will be chosen from the remaining possibilities for which the length L of the edges are as small as possible (since H is fixed). This means that for situation of Figure 12 b) the splitting of Figure 13 b) will be chosen. Similarly for situations c) and d). For situation a) of we impose configuration I) from Figure 3, in order to avoid creation of unnecessarily small triangles. Note how the dotted edges are forced to lie almost parallel to the contour while its endpoints are attracted towards the discontinuity like can be seen from Corollary 5.3. The latter is thanks to the *adapted normal offset decomposition*, the former is thanks to the *adaptive splitting*, which attracts edges towards the one dimensional line singularities.

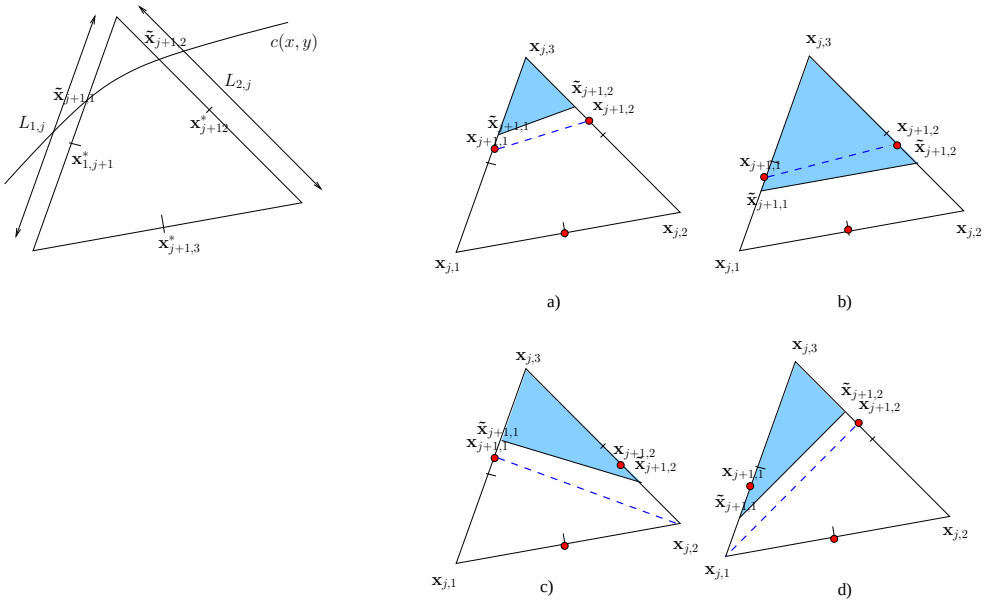


Figure 12: *The Figure depicts the five situations in which a contour (separating the two regions with different functions values) can cut a triangle with respect to its piercing points. (Remember that the piercing point always lies between the midpoint of the edge and the point where the edge is cut by the contour). The blue areas define the region Ω' in which $f(x,y)_{\Omega'} = H$, and all points of the white area $\Omega \setminus \Omega'$ having function value 0. The red dots represent the piercing points, the short stripes the midpoints. Interchanging the function value of the blue and white areas of one of the above pictures, just brings you in another depicted situation, i.e. you go from situation d) to e) and visa versa while the rest of the situations are reflexive with the transformation applied. So, we just have to analyze 5 situations.*

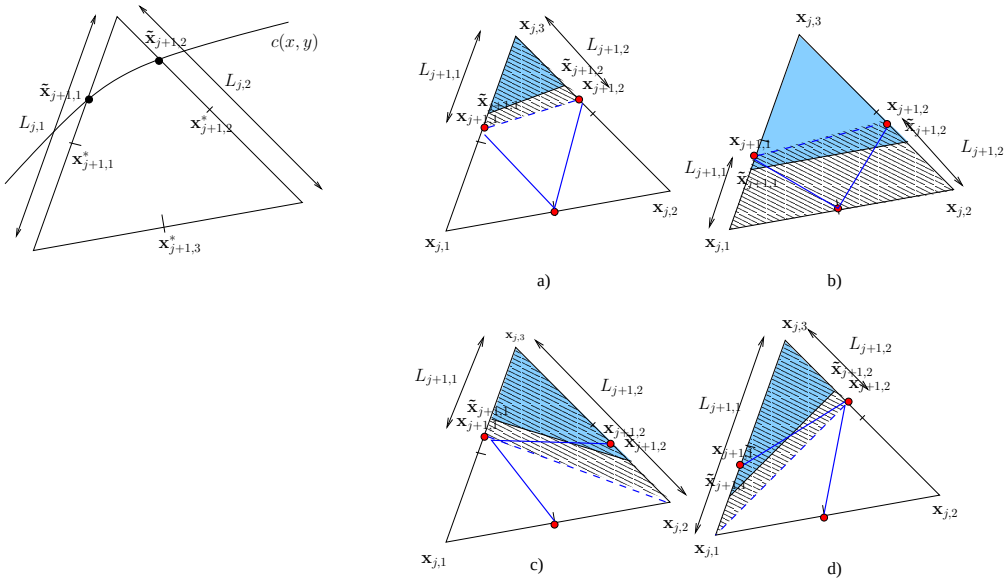


Figure 13: *Starting from Figure 12, the best splittings amongst those from figure 4 are being depicted for each situation. The cross hatched areas represent the area above which the approximating mesh does not coincide with the surface of the 2D function.*

Proposition 5.6. *Assume all situations depicted in Figure 12 are equally likely to occur, then the expected number of triangles cut by the contour, $\#E^{\not\triangle}_{j+1}$, in the next level $j+1$ is:*

$$\#E^{\not\triangle}_{j+1} = 2\#E^{\not\triangle}_{j} \tag{22}$$

Proof. Under the assumption and considering Table 1 for the adaptive approach:

$$\#E^{\not\triangle}_{j+1} = \sum_{k=0}^{\#E^{\not\triangle}_{j}-1} \frac{1}{4}\left(1+3+2+2\right) = 2\#E^{\not\triangle}_{j}$$

□

Mark that using Table 1 using a fixed splitting method, previous expression would become

$$\#E^{\not\triangle}_{j+1} = \sum_{k=0}^{\#E^{\not\triangle}_{j}-1} \frac{1}{4}\left(1+3+3+3\right) = 5/2\#E^{\not\triangle}_{j}$$

in the case of a fixed partition.

Lemma 5.1. *Still under the assumption of Equation (18), and under the condition that there exists a value ϵ sufficiently small, for which*

$$1-\epsilon \leq \frac{L_{j,1}}{L_{j,2}} \leq 1+\epsilon.$$

The area reduction is:

$$E\left(\frac{A_{j+1}(\not\triangle)}{A_j(\not\triangle)}\right) \leq \frac{1}{2}\left(1-\left(\eta_0 2^j 2^{(C(4^j-1))}\right)^2\right) \tag{23}$$

Proof. We set $L_j = \dfrac{L_{j,1}+L_{j,2}}{2}$ and $L_{j+1} = \dfrac{L_{j+1,1}+L_{j+1,2}}{2}$. Referring to the situation depicted in Figure 13 it follows that, for ϵ sufficiently small:

a) $A_{j+1}(\not\triangle) \simeq A_j(\not\triangle)\left(\dfrac{L_{j+1}}{L_j}\right)^2$

b) $A_{j+1}(\not\triangle) \simeq A_j(\not\triangle)\left(1-\left(\dfrac{L_j-L_{j+1}}{L_j}\right)^2\right)$

c) $A_{j+1}(\not\triangle) \simeq A_j(\not\triangle)\dfrac{L_{j+1}}{L_j}$

d) $A_{j+1}(\not\triangle) \simeq A_j(\not\triangle)\dfrac{L_{j+1}}{L_j}$

$$
\begin{aligned}
E\left(\frac{A_{j+1}(\not\triangle)}{A_j(\not\triangle)}\right) &= \frac{1}{A_j}(\not\triangle)\left(\frac{1}{4}A_j(\not\triangle).\left(\left(\frac{L_{j+1}}{L_j}\right)^2+\left(1-\left(\frac{L_j-L_{j+1}}{L_j}\right)^2\right)+2\frac{L_{j+1}}{L_j}\right)\right)\\
&= \frac{L_{j+1}}{L_j}\\
&\quad \text{from Equation (19)}\\
&= \frac{1}{2}\left(1-\eta_j^2\right)\\
&\quad \text{from Equation (21)}\\
&\leq \frac{1}{2}\left(1-\left(\eta_0 2^j 2^{(C(4^j-1))}\right)^2\right)
\end{aligned}
$$

□

If all areas have the same height, i.e. $H_{j+1} = H_j = H$ the L_1-error decays like:

$$
\begin{aligned}
E\left(\frac{\|f-f_{j+1}\|_{L_{j+1}}}{\|f-f_j\|_{L_j}}\right) &= E\left(V_{j+1}(\not\triangle)/V_j(\not\triangle)\right)\\
&= \left(\frac{\frac{1}{3}HA_{j+1}(\not\triangle)}{\frac{1}{3}HA_j(\not\triangle)}\right)\\
&= \frac{1}{2}\left(1-\left(\eta_0 2^j 2^{(C(4^j-1))}\right)^2\right) \qquad (24)
\end{aligned}
$$

5.2.2 Subcase I): $c(\mathbf{x_{j,1}}) \equiv 0 \ \& \ c(\mathbf{x_{j,2}}) \equiv 0$

Figure 15 represents the situation where both edges cut by the contour have their normal rays pierced through the discontinuity at $p_{j+1,1}$ and $p_{j+1,2}$. The Figure shows the interconnection as the one depicted in picture I) of Figure 4, this however does not have to be the case, as long as the edge $e = (p_{j+1,1}, p_{j+1,2})$ is present in the splitting, leaving some other possibilities open. In this paper we decide to take the interconnection from the one that gives the smallest L_1-error letting the vertical offsets play part into the approximating mesh. It is the first time vertical offsets are introduced in the analysis, because piercing through discontinuities vertical offsets of considerable magnitude come into play. In the previous section, the normal ray never pierced through a discontinuity, such that the piercing point always took on a value of one of the separated regions, i.e. $z = H$ or $z = 0$. The extra vertical offset needed to exactly represent the function values are large only when pierced trough a discontinuity.

In the previous section, the volumes between the real and the interpolating approximation on each triangle had a height either zero or H. In this step,however, the heights are reduced by early cutoff of normal rays by the discontinuity before hitting the ceiling or floor of the Horizon image. Accordingly, we have a decay with respect to the height H_j, in contrast with Equation (24). The error is situated only at one triangle as can be seen from Figure 15. The obtained error reduction is:

$$
E\left(\frac{\|f-\mathcal{M}_{j+1}\|_{L_1}}{\|f-\mathcal{M}_j\|_{L_1}}\right) = 2^{-2} \qquad (25)
$$

The main benefit is that in the next steps no contour cutting is finding place, i.e. $A_{j+1}(\not\triangle) = 0$. In the next step there is a chance that two piercing points are found on the same location $\mathbf{x}$, i.e. pierced through

the same *vertical* edge. Then , you simply keep the piercing point p that lies closest to the real function value $f(\mathbf{x})$. The quadtree structure reduces to a binary tree since those triangles are cut in two. For images $\in \mathcal{PS}$ we recommend this approach, producing more horizontal edges having steeper normal rays. In the case of images $\in \mathcal{H}$ it actually does not matter since the function value was already obtained by a previous piercing point and its vertical offset.

Remark 5.1. *In the discrete setting there are always two candidate pixels which can serve as the location of the piercing pixel i.e., the one right after the jump detection or the one in front. Equation (14) as discussed in Problem 3 (Section 4) becomes important in the way that piercing point are always found at the same side of the contour, and combined with their vertical offsets always lead to the error reduction as mentioned in Equation 25.*

5.2.3 Approximation results

We obtained experimental results indicating the rate of approximation of the proposed method using a simple pruning algorithm. Starting from the full decomposition tree, we pruned the parents children that introduced the least decay of the L_1 error norm. Next to the measurement of the total L_1 error norm, the number of non-zero offsets was counted at each step. Counterintuitive to the tree structured organization of the triangles, the number of normal offsets is directly related to the number of edges present in the mesh. In fact every edge which is not a leaf has exactly one normal offset. An edge which is part of a leaf can have one or zero normal offsets depending on wether the adjacent triangle (with the edge in common) is further refined or not. Apart from normal offsets (indices) we have to take the number of vertical offsets into account. These vertical offsets play an important role at the stage where normal rays are piercing through the line discontinuities. As we can see from Figure 17(d), the rate of approximation changes goes in two stages. In the first stage most of the piercing points pierce through the surface . In the second stage, at the moment piercing points are found along the contour,the rate of approximation increases. The optimal rate of approximation is limited by the polyline approximation of the contour. This makes that for a smooth $\mathcal{C}^2$ contour an approximation error is made of the order $O(L_j^{-3})$ at each piece of the polyline, with L_j the length of that piece (i.e. a triangle edge). This length decreases like 2^{-j}. The number of pieces in the polyline, however , increases like 2^j. Also the number of non-zero offsets n increases with the number of pieces (2^j). This all adds up to a theoretical error norm decay $\epsilon_n = O(n^{-1})$.

6 Conclusions and Further Research

The proposed method solves major issues concerning the compression of discrete images. This paper has extensively studied the properties of normal offsets in the context of rather artificial images (images being part of $\mathcal{H}$). Figure 16 once again shows a compressed Horizon class image together with its domain partitioning. As for real life images having smoothly colored regions separated by smooth contours, we experimented with the well-known 'peppers' image. Although the algorithm as is was built solely for Horizon class images its results indicate a good performance on images $\in \mathcal{PS}$. An example of the algorithm applied to the 'peppers' image is depicted in Figure 18. Further research will adapt some parts of the algorithm to improve its performance on those images. In a forthcoming paper, the vertical offsets will be given more attention and will be subjected to an postprocessing optimization procedure. Although we already anticipated stability issues as explained in Section 4 (Problem 4), the behavior in presence of noise – or more broadly speaking stability issues – of the nonlinear normal offset algorithm has to be further analyzed.

7 Acknowledgements

This work is funded by the FWO (Flemish Fund for Scientific Research), being part of the SMID project G.0431.05 (Stability of Multiscale Transforms on Irregular Data).

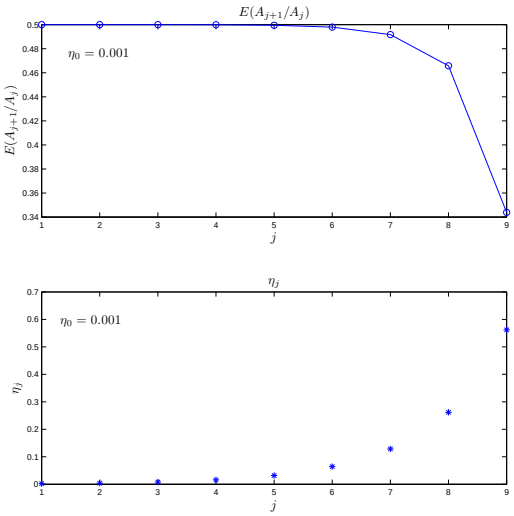


Figure 14: *(Top)The expected decay of the ratio $A_{j+1}(\not\triangle)/A_j(\not\triangle)$ is depicted. (Bottom) Gives the corresponding η_j value, for the picture on top. Note that the theory of this section only holds for $\eta_j < 1$.*

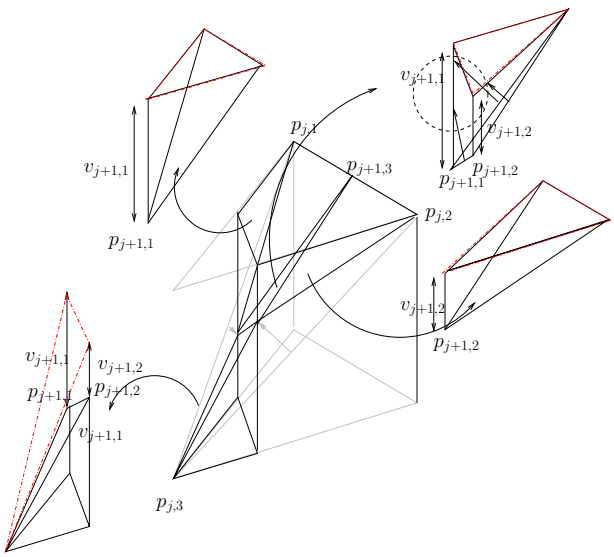
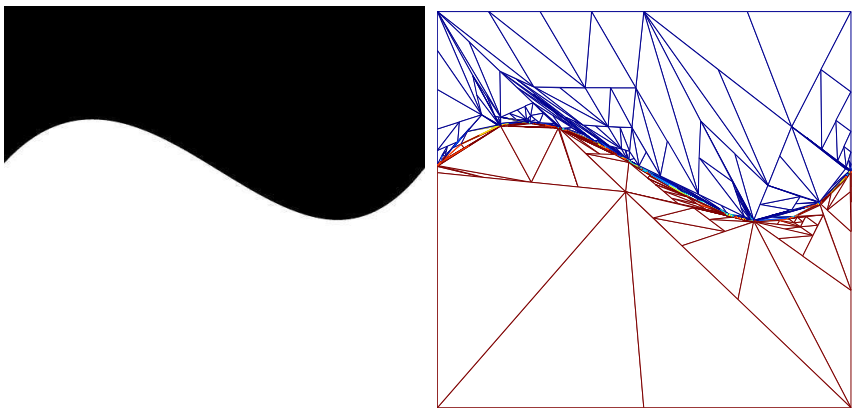


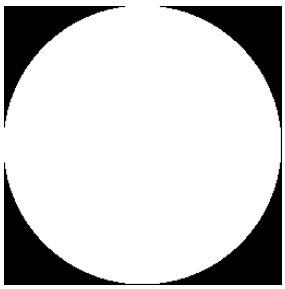
Figure 15: *The above picture represents a case where both edges cut by the contour have there normal rays pierced through the discontinuity. In the next step there is a chance that two piercing points are found on the same location. Then , you simply keep the one that lies closest to the real function value.*



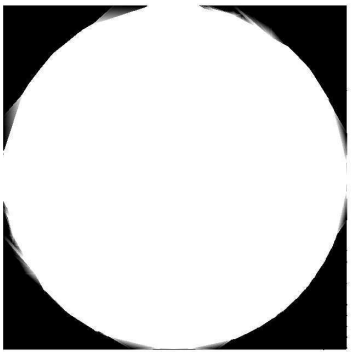
(a) Original image $\in \mathcal{H}$.

(b) Adaptive partition obtained after seven normal offsets iterations by refining all triangles but the ones which have already minor contribution to $\epsilon_{L_1}(\mathcal{M}_j)$ (less then a certain threshold).

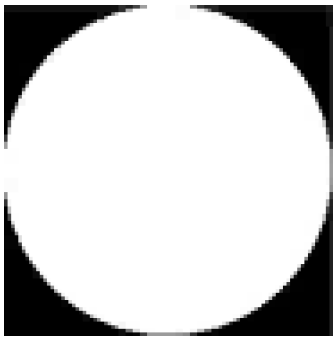
Figure 16: *Result of an Horizon class image together with its domain partitioning.*



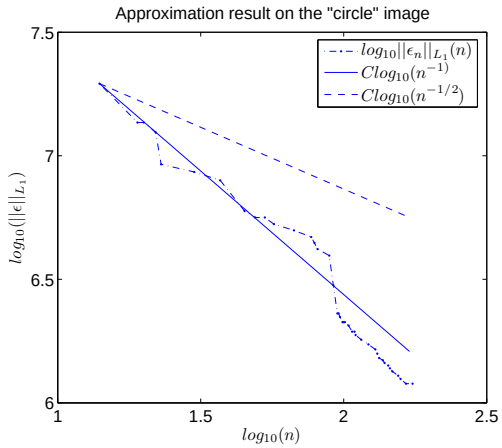
(a) Test picture $\in \mathcal{H}$ with resolution 512×512.



(b) The contour is approximated by a piecewise linear polyline, in contrast with tensor product wavelets where the contour is approximated by a piecewise constant curve.



(c) Result of a 1042-terms approximation by tensor product wavelets. The blockyness of the piecewise constant approximation of the contour is clearly visible.



(d) Plot of the error $\epsilon_n(L_1)$ versus the used number of offsets, together with a trend line of the optimal approximation rate $\alpha = 1$ and the theoretical approximation rate using a tensor product wavelet method on the Horizon class images $\alpha = 1/2$.

Figure 17: *Results for an Horizon class image*



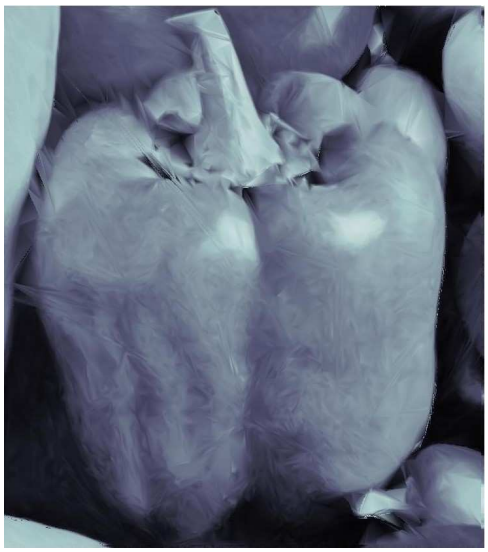
(a) Original image named 'peppers'.



(b) Representation of $\mathcal{M}_7$.



(c) Selected cut from image a), where mainly one of the objects ('pepper') is presented.



(d) Representation of $\mathcal{M}_7$. We see that when two contours come close to each other the algorithm is attracted to both of them resulting in what you can see at the left side of the picture. Having only one contour like at the right side of the image, the algorithm is not distracted and does a great job in approximating the contour.

Figure 18: *Results for the 'peppers' image.*

References

[1] J.E. Bresenham. Algorithms for computer control of a digital plotter. *IBM Systems Journals*, 4(1):25–30, 1965.

[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] M. N. Do and M. Vetterli. Contourlets. In G. Welland, editor, *Beyond Wavelets*, pages 83–107. Academic Press, 2003.

[5] D. L. Donoho. Wedgelets: Nearly minimax estimation of edges. *Annals of Statistics*, 27(3):859–897, 1999.

[6] D. L. Donoho. Orthonormal ridgelets and linear singularities. *SIAM J. Math. Anal.*, 31:1062–1099, 2000.

[7] Leila De Floriani and Enrico Puppo. Hierarchical triangulation for multiresolution surface description. *ACM Trans. Graph.*, 14(4):363–411, 1995.

[8] I. Guskov, K. Vidimee, W. Sweldens, and P. Schröder. Normal meshes. In *SIGGRAPH 2000 Conference Proceedings*, 2000.

[9] O. Runborg I. Daubechies and W. Sweldens. Normal multiresolution approximation of curves. *Constructive Approximation :20*, pages pp.399–463, 2004.

[10] M. Jansen, R. Baraniuk, and S. Lavu. Multiscale approximation of piecewise smooth two-dimensional functions using normal triangulated meshes. *Appl. Comp. Harm. Anal.*, 2005.

[11] E. Le Pennec and S. Mallat. Sparse geometrical image representations with bandelets. *submitted*, 2003.

[12] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:pp. 379–423 and 623–656, July and October, 1948.

[13] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2):186–200, 1996.