

# Normal Mesh based Geometrical Image Compression

W. Van Aerschot <sup>a,\*</sup>, M. Jansen <sup>b</sup> and A. Bultheel <sup>a</sup>

<sup>a</sup>*K.U.Leuven, Department of Computer Science, Celestijnenlaan 200A, 3000 Leuven, Belgium*

<sup>b</sup>*K.U.Leuven, Department of Mathematics, Celestijnenlaan 200B, 3000 Leuven, Belgium*

---

## Abstract

Recently the performance of nonlinear transforms have been given a lot of attention to overcome the suboptimal  $n$ -terms approximation power of tensor product wavelet methods on higher dimensions. The suboptimal performance prevails when the latter are used for a sparse representation of functions consisting of smoothly varying areas separated by smooth contours. This paper introduces a method creating normal meshes with *nonsubdivision* connectivity to approximate the nonsmoothness of such images efficiently. From a domain decomposition viewpoint, the method is a triangulation refinement method preserving contours. The transform is nonlinear as it depends on the actual image. This paper proposes an normal offset based compression algorithm for digital images. The discretisation causes the transform to become redundant. We further propose a model to encode the obtained coefficients. We show promising rate distortion curves and compare the results with the JPEG2000 encoder.

*Key words:* normal multiresolution mesh, image compression, wavelets

---

## 1 Introduction

Normal meshes have been introduced as a method for surface representation in computer graphics applications. They allow sparse representations of surfaces approximating enormous amounts of data coming from scanning smooth

---

\* Corresponding author.

*Email addresses:* ward.vanaerschot@cs.kuleuven.be (W. Van Aerschot), maarten.jansen@wis.kuleuven.be (M. Jansen), adhemar.bultheel@cs.kuleuven.be (and A. Bultheel).

objects [2,10,12,14]. Each vertex can be represented by one coefficient in a local frame instead of its usual three coordinates. This property makes normal meshes also suited for mesh compression [14]. A more theoretical study investigates the approximation of smooth curves with polylines [4].

This paper investigates the application of normal meshes to compress images whose content is dominated by contours. This kind of images will be referred to as *geometrical* images. For this type of images, transform coders using nonredundant bases fall short when creating compact high resolution representations. The highest achievable  $n$ -term approximation rate using a wavelet transform combined with a nonlinear thresholding equals  $\sigma_{L_2} = O(n^{-\frac{1}{2}})$  [6]. Therefore, transform coders like JPEG2000 will perform suboptimally compared to recently developed schemes capable of sparsely representing line discontinuities.

These schemes can be classified into two categories. On the one hand, redundant linear transforms, like directional frames, combined with a nonlinear  $n$ -term selection e.g. contourlets [7], curvelets [3]. On the other hand, we have *highly* nonlinear schemes where also the transform is nonlinear. Examples are bandelets [13], geometric wavelets [5], dictionary approaches and so on. The nonlinearity of the transform partitions the image domain into parts of different sizes and shapes. The small parts correspond to image regions that contain much information and the large parts to smooth image regions. The normal offset scheme belongs to this category.

Images represented as functions on a 2-d plane can also be seen as surfaces in  $\mathbb{R}^3$ , which generally are approximated by triangular meshes. There are many papers involving normal mesh compression [8–10,12,14]. Despite the similarities between surface and image compression we cannot take ‘off the shelf’ normal mesh encoders and apply them to the surface equivalents of a geometrical image. The first major difference concerns the smoothness properties of the target surfaces. While surfaces in computer graphics are smooth, geometrical images are non-smooth: the edges are the information carrying feature. A second major difference – and the actual reason for using a normal offset transform in an image compression algorithm– is the exploitation of normal approximation properties in order to compress. The reason for using normal offsets in surface rendering applications is that they contain almost all geometrical information [10]. Thanks to the minor contribution of the parametrical information, semi-regular meshes, demanding almost no connectivity information can be utilised. As such, the surface can be approximated by a mesh where each point is defined using a single scalar. This leads to a bonus compression factor of three. Image grey levels are not triple coordinates in  $\mathbb{R}^3$ , but single values on a given, regular lattice, so there is no bonus to gain in the image case. Irregular meshes have to be used to preserve contours. The fundamental reason for using normal offsets in image processing is the edge

locating property of the normal search direction. The application of normal offsets to images leads to a different implementation.

A first attempt to use normal mesh techniques for image approximation was made in [11]. Gray scale images are treated as two dimensional functions dominated by geometric structures comparable with terrain models used in geographical information systems. The authors achieve an  $n$ -term approximation rate that is twice as good as wavelet approximations on the studied images. They do not address compression, but since approximation and compression are tightly related, their results indicate that the normal offset method should be considered for the development of rate-distortion efficient image encoders. Also results given in an earlier paper [18] indicate that for images of geometrical nature the normal offset decomposition is a promising compression technique. The multi-resolution model used in [11] is non nested. Therefore additional information needs to be stored in order to encode the complex dependency relations between consecutive resolution levels. They do not incorporate a contour preserving triangulation method which explains the rather poor visual results of the presented transform.

We propose a nested multi-resolution model with a more adequate definition of the normal direction w.r.t. compression of non-smooth functions instead of smooth surfaces. We use *local* mesh refinements with additional adaptivity to preserve the approximation strength. The use of local mesh refinements results in tree-structured dependency relations between successive approximations. The refinements are simple triangle splits that can be encoded in a straightforward manner.

The paper is outlined as follows. In Section 2 the proposed normal mesh algorithm is given in detail. This nonlinear transform forms the main component of the encoder. It produces sparse representations of geometrical images. The efficient encoding of the wavelet coefficients (normal offsets) is done by a model based entropy coder and will be explained in Section 3. Finally Section 4 shows rate-distortion curves of the proposed encoder and compares the results with the state of the art JPEG2000 encoder.

## 2 The normal offset scheme

Given an interpolating triangular mesh  $\mathcal{M}_0$  with vertices  $\mathcal{V}_0$ , edges  $\mathcal{E}_0$  and triangles  $\mathcal{T}_0$ , the normal offset scheme consists of an iterative application of three steps. The first step, the *prediction* step, constructs additional vertices – prediction points – as linear combinations of surrounding mesh points. The second step, the *correction* step, constructs piercing points as the intersection of rays normal to the coarser mesh, going through those prediction points and

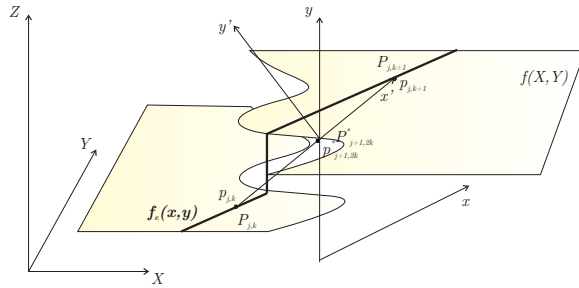


Fig. 1. Global coordinate system  $XYZ$  together with local coordinate frames  $xy$  and  $x'y'$ .

the image surface. The third step, the *interconnection* step, adds all piercing points to the set of vertices  $\mathcal{V}_j$  forming  $\mathcal{V}_{j+1}$ . The corresponding mesh  $\mathcal{M}_{j+1}$  with edges  $\mathcal{E}_{j+1}$  and triangles  $\mathcal{T}_{j+1}$  is constructed by a triangulation of  $\mathcal{V}_{j+1}$ . These steps are repeated, creating meshes at different resolution levels  $j$  until a certain stopping criterion is met.

We emphasise some major differences where our method distinguishes itself the method proposed in [11]. We restrict the rays emanating from the prediction point to lie in a plane perpendicular to the  $XY$  plane, still forming a right angle ( $\angle = \frac{\pi}{2}$ ) with an edge of the coarser mesh. Also the choice of the *triangulation method* is crucial for approximation and compression performance. Despite the theoretical approximation rate of  $O(n^{-1})$  using a Delaunay retriangulation it is not particularly suited for compression. The edges defining the local frames for the piercing points can disappear in higher resolution levels. This requires a lot of bookkeeping when we want to reconstruct the compressed image from thresholded and quantised coefficients. Even vertex connectivity can change when the triangulation method is sensitive to perturbations in the vertex positions due to a quantisation step on the normal coefficients. This further complicates the compression stage. Furthermore a lot of exception handling has to be done to avoid triangle fold overs since the normal direction does not guaranty a proper parametrisation. We therefore opt for a local triangulation scheme equipped with a certain amount of flexibility to preserve contours. The level-to-level mesh refinement relations can be captured by tree structures.

## 2.1 The algorithm

We now give a specific implementation of each step the normal offset *transformation* scheme:

- (a) **Prediction:** In what follows, we use linear interpolation for the prediction step. Suppose the image is given as  $S_f := (X, Y, f(X, Y))$ . In order to obtain an hierarchical edge refinement scheme we consider the subspace defined as the vertical plane ( $\perp XY$ ) containing edge  $e_{j,k} \in \mathcal{E}_j$ . The part

of the image contained in the vertical plane is denoted by  $f_e$ . We attach a  $2d$  coordinate system to this subspace with the  $x$ -axis in the  $XY$ -plane and the  $y$ -axis parallel w.r.t. the  $Z$ -axis (see Figure 1). The prediction point  $p_{j+1,2k+1}^* = (x_{j+1,2k+1}^*, y_{j+1,2k+1}^*)$  on location  $2k + 1$  and resolution level  $j + 1$  is defined as:

$$p_{j+1,2k+1}^* := \frac{p_{j,k} + p_{j,k+1}}{2}$$

- (b) **Correction:** Corrects the location of the new points by adding a vector in the direction normal to the coarse mesh. The normals  $\vec{n}_{j,k}$ , on the coarse mesh such that  $\|\vec{n}_{j,k}\| = 1$  are expressed as

$$\vec{n}_{j,k} = \begin{bmatrix} n_{[x]j,k} \\ n_{[y]j,k} \end{bmatrix} := [-(y_{j,k+1} - y_{j,k}), x_{j,k+1} - x_{j,k}] / \|p_{j,k+1} - p_{j,k}\|,$$

and represent the direction of the perpendicular bisector on  $e_{(p_{j,k+1}, p_{j,k})}$ . The normal ray  $r_{j+1,2k+1}$  going through  $p_{j+1,2k+1}^*$  is defined as:

$$r_{j+1,2k+1}(\gamma) := p_{j+1,2k+1}^* + \gamma \vec{n}_{j,k}$$

For  $f_e \in \mathcal{C}$ , with  $\mathcal{C}$  the class of continuous functions, the correction step calculates

$$\gamma_{j+1,2k+1} := \min \{ \gamma \mid r_{j+1,2k+1}(\gamma) = [x, f_e(x)] \}. \quad (2.1)$$

The minimum value of  $\gamma$  over a set is taken, since in general the normal ray can pierce  $f_e$  more than once.

For  $f_e \notin \mathcal{C}$  and  $r$  parameterised as  $r_{j+1,2k+1}(\gamma) = (x_{j+1,2k+1}(\gamma), y_{j+1,2k+1}(\gamma))$ , Eq. (2.1) takes the following form:

$$\begin{aligned} \gamma_{j+1,2k+1} &:= \min \left\{ \gamma \mid \text{sign} \left( \lim_{x_{j+1,2k+1}(\gamma) \rightarrow x^+} y_{j+1,2k+1}(\gamma) - f_e(x) \right) \right. \\ &= \left. - \text{sign} \left( \lim_{x_{j+1,2k+1}(\gamma) \rightarrow x^-} y_{j+1,2k+1}(\gamma) - f_e(x) \right) \right\}. \end{aligned} \quad (2.2)$$

As such we always obtain a subdivision scheme i.e.,  $x_{j,k} \leq x_{j+1,2k+1} \leq x_{j,k+1}$ . Setting  $x_{j+1,2k} := x_{j,k}$  we obtain a monotonically increasing sequence  $\mathbf{x}_j = \{x_{j,k}, x_{j,k+1}, x_{j,k+2}, \dots\}$  on each edge. A new piercing point  $p_{j+1,2k+1}$  is inserted between  $p_{j,k}$  and  $p_{j,k+1}$  in order to form the sequence

$\mathbf{p}_{j+1}$ :

$$\begin{aligned} p_{j+1,2k} &\leftarrow p_{j,k} \\ p_{j+1,2k+1} &\leftarrow r_{j+1,2k+1}(\gamma_{j+1,2k+1}). \\ p_{j+1,2k+2} &\leftarrow p_{j,k+1} \end{aligned}$$

Each slice  $f_e$  created by the previous step is gradually approximated by a polyline (a continuous curve composed of several line segments) through the sequence  $\mathbf{p}_j = \{p_{j,0}, \dots, p_{j,k} \dots, p_{j,2^j+1}\}$ .

- (c) **Interconnection:** Each coarse triangle is split into 4 disjunct subtriangles whose projection on the  $XY$ -plane form a planar graph. The inner edges of the tessellation are formed from couples from the set of vertices of the coarse triangle and the piercing points. The choice of these inner edges yields four possible triangle splits. The split minimising an error metric is included into the finer resolution mesh. We will use the  $L_2$ -error metric between the original surface  $S_f$  and the approximating mesh  $\mathcal{M}_j$  in the remainder of the paper.

## 2.2 Digital setting

Consider a digital image obtained from sampling an analogue grayscale image. A *digital* grayscale image is defined as a regular tessellation of *pixels*, disjunct rectangular areas having a gray value. The gray values attached to a pixel take on a discrete value. In the digital setting two pixels together with the collection of adjacent pixels connecting both of them define a *discrete* edge. The collection of pixels is generated by a line rasterisation algorithm  $\mathcal{B}^1$  taking the two end pixel locations as input giving back an array of in between pixel locations. The number of pixels generated by  $\mathcal{B}(P_{j,k}, P_{j,k+1})$  is represented by  $L_{j,k}$ . We use the notation  $\mathcal{B}_{[X]}(P_{j,k}, P_{j,k+1})_i$ ,  $i \in [0 \dots L_{j,k} - 1]$  to indicate the  $X$ -value of the  $i$ th element. The pixels at both ends of a digital edge are the digital counterparts of a vertex. As such discrete edges have a nonzero area, just as discrete points (namely pixels). Accordingly three discrete edges – having pairwise an end point in common– together with the pixels residing in the interior define a *discrete* triangle.

In this paper we approximate a *digital* image by a *discrete mesh* which is defined as the triple of the set of discrete points, edges and triangles. For the sake of brevity we will drop the word ‘discrete’ in the remainder of this paper.

The definitions used in Section 2.1 have to be adjusted towards the digital setting. In the digital setting where the pixels  $(X, Y, f(X, Y)) \in \mathbb{N} \times \mathbb{N} \times \mathbb{R}$ ,

---

<sup>1</sup> In this paper we take for  $\mathcal{B}$  Bresenham’s line rasterisation algorithm.

the prediction point expressed in the global coordinate system is defined as:

$$P_{j+1,k+1}^* = \begin{bmatrix} \mathcal{B}_{[X]}(P_{j,k}, P_{j,k+1})_{\lfloor L_{j,k}/2 \rfloor} \\ \mathcal{B}_{[Y]}(P_{j,k}, P_{j,k+1})_{\lfloor L_{j,k}/2 \rfloor} \\ \frac{y_{j,k} + y_{j,k+1}}{2} \end{bmatrix}$$

such that also  $P_{j,k}^* \in \mathbb{N} \times \mathbb{N} \times \mathbb{R}$ . The definition of the normal ray in the previous section is slightly adapted for the digital setting.

**Definition 2.1 (normal ray)** *The digital normal ray  $\bar{r}_{j+1,2k+1}(\gamma)$  expressed in the coordinate system  $xy$  through  $e_{j,k}$  is defined as:*

$$\bar{r}_{j+1,2k+1}(\gamma) = \text{sign}(\gamma) \left( \begin{bmatrix} 0 \\ \frac{y_{j,k} + y_{j,k+1}}{2} \end{bmatrix} + d_{j,k}(\gamma) \begin{bmatrix} 1 \\ \frac{n_{[y]_{j,k}}}{n_{[x]_{j,k}}} \end{bmatrix} \right), \gamma = -\lfloor L_{j,k}/2 \rfloor \dots \lceil L_{j,k}/2 \rceil \quad (213)$$

and  $d_{j,k}(\gamma) = \left\| \left( \mathcal{B}(P_{j,k}, P_{j,k+1})_{\gamma}, \mathcal{B}(P_{j,k}, P_{j,k+1})_{\lfloor L_{j,k}/2 \rfloor} \right) \right\|_{L_2}$  the distance between the midpixel of  $e_{j,k}$  and the pixel with index  $\gamma$ .

**Definition 2.2 (normal index)** *We define the normal index  $i_{j,k}$  as the  $\gamma$  (Eq. (2.3)) for which Eq. (2.2) holds. The normal index corresponds to the index of the pixel on the rasterised edge above which the image is pierced.*

The integer valued *normal indices* will be stored instead of the real valued normal offsets  $\gamma$ . Next section further investigates normal indices.

Contrary to the continuous setting the normal offset algorithm for the digital setting is unlikely to exactly pinpoint a sample as seen from Eq. (2.2). Due to discretisation in most of the cases  $y_{j,k}$ , the  $y$ -value of the piercing point, will differ from the exact function value  $f_e(x_{j,k})$ . The  $y$ -values that can be taken on by the piercing points are separated by a distance  $\Delta y = s^{-1}$ , with  $s \in \mathbb{R}$  the slope of the normal ray  $\bar{r}_{j,k}(\gamma)$ . To ensure perfect reconstruction a vertical offset  $v_{j,k}$  is introduced.

**Definition 2.3 (vertical offset)** *The vertical offset is the difference between the exact function value and the value of the digital ray (Definition (2.1)) at  $i_{j,k}$ :*

$$v_{j,k} = f_e(x_{j,k}) - \bar{r}_{j,k}(i_{j,k}). \quad (2.4)$$

This extra offset caused by the discretisation process makes the transform redundant.

### 3 Compression

The previous section stated the transform part of the encoder. This section discusses how the obtained coefficients are encoded. A previous paper [18] explains data structures used by the normal mesh algorithm and discussed the encoding of the hierarchical mesh structure to a serialised bitstream. This paper introduces a probability model for the normal indices  $i_{j,k}$  and vertical offsets  $v_{j,k}$  in order to reduce the overall bitrate given a target signal to noise ratio. Apart from the local function behavior, the PMF (Probability Mass Function) of  $i_{j,k}$  depends on orientation of the edge  $e_{j,k}$  w.r.t. the  $XY$ -plane. Once we have a sound model for the PMF in terms of  $H_{j,k} := |Z_{j,k+1} - Z_{j,k}|$  and  $l_{j,k} := \|(X_{j,k+1}, Y_{j,k+1}), (X_{j,k}, Y_{j,k})\|$  for those indices  $i_{j,k}$ , we can apply an entropy-coder to reduce the expected bitrate.

#### 3.1 Model for the vertical offsets

The detail information or the high-frequency part of the input signal is gathered into vertical offsets. According to [15] and confirmed by the experiments shown in Figure 2, the vertical offsets  $v$ , can be modelled by a two-parametric zero inflated geometrical distribution  $ZID(p, \lambda)$  (see [1] and references therein) given by:

$$v \sim f_V(v) = p\delta(v) + (1 - p)\lambda/2(1 - \lambda)^v. \quad (3.1)$$

In the next paragraph we discuss how to obtain the parameters  $p$  and  $\lambda$  using a most-likelihood estimate (MLE) method.

##### 3.1.1 Zero inflated distribution: parameter estimation

Given a parametric distribution  $\pi(v|\lambda)$ . We define the associated zero-inflated distribution to have:

$$P(V = 0|p, \lambda) = p + (1 - p)\pi(0|\lambda) \quad (3.2)$$

$$P(V = v|p, \lambda) = (1 - p)\pi(v|\lambda) \quad (3.3)$$

with  $p$  the chance that the hidden state  $Z$  finds itself in the  $\delta$  distribution. We have the following properties:

$$\begin{aligned} E(V|p, \lambda) &= (1 - p)E_\pi(V|\lambda) \\ \text{Var}(V|p, \lambda) &= p(1 - p)E_\pi(V|\lambda)^2 + (1 - p)\text{Var}_\pi(V|\lambda) \end{aligned}$$



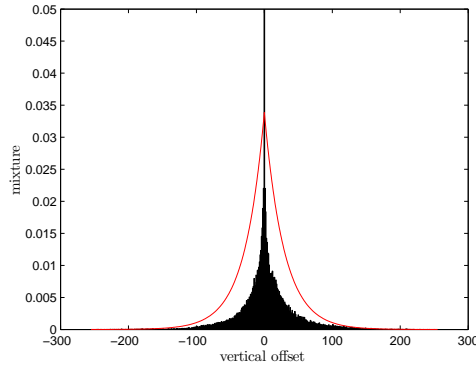


Fig. 2. Histogram of the vertical offsets coming from a large test set of geometrical images. The figure suggests a zero inflated double geometrical distribution to model the statistics of the vertical offsets. The parameters produced by the MLE procedure are  $p = 0.3955$  and  $\lambda = 0.0339$ .

For a sample of size  $n$  ( $v_i$  given) the full likelihood is given by:

$$\begin{aligned}
 L(p, \lambda|V, Z) &= \prod_{i=1}^n P(V = v_i|Z_i = z_i)P(Z_i = z_i) \\
 &= \prod_{i=1}^n p^{z_i}((1-p)\pi(v|\lambda))^{1-z_i} \\
 &= \prod_{v_i > 0} ((1-p)\pi(v_i|\lambda)) \prod_{v_i = 0} p^{z_i}((1-p)\pi(0|\lambda))^{1-z_i} \quad (3.4)
 \end{aligned}$$

The log-likelihood is given by:

$$\begin{aligned}
 \log L(p, \lambda|V) &= N_0 \log(p + (1-p)\pi(0|\lambda)) \\
 &\quad + (n - N_0) \log(1-p) + \sum_{v_i > 0} \log(\pi(v_i|\lambda)) \quad (3.5)
 \end{aligned}$$

where  $N_0$  is the number of zero occurrences.

Suppose we model the samples with a zero inflated geometrical distribution, i.e.,  $\pi(v|\lambda) \sim \lambda(1-\lambda)^v$ ,  $v \in \mathbb{N}$ . Equation (3.5) becomes:

$$\begin{aligned}
 \log L(p, \lambda|V) &= N_0 \log(p + (1-p)\lambda) \\
 &\quad + (n - N_0) (\log(1-p) + \log \lambda) + \log(1-\lambda)\bar{V} \quad (3.6)
 \end{aligned}$$

with  $\bar{V} = E_{v_i > 0} [v_i]$ , the mean of the values  $v_i > 0$ . We have to maximize Eq. (3.6) under the constraints  $0 \leq p \leq 1$  and  $0 \leq \lambda \leq 1$ . Note that Eq. (3.6) is concave within the feasible region. This can be translated into an ICP (inequality constrained problem), which can be solved by a Newton minimization procedure. The derivatives of  $\log L(p, \lambda|V)$  are:

$$\begin{aligned}\frac{\partial \log L(p, \lambda|V)}{\partial p} &= N_0 \frac{(1 - \pi(0|\lambda))}{p + (1 - p)\pi(0|\lambda)} - (n - N_0) \frac{1}{1 - p} \\ \frac{\partial \log L(p, \lambda|V)}{\partial \lambda} &= N_0 \frac{\pi'(0|\lambda)(1 - p)}{p + (1 - p)\pi(0|\lambda)} + \sum_{v_i > 0} \frac{\pi'(v_i|\lambda)}{\pi(v_i|\lambda)}\end{aligned}\quad (3.7)$$

with  $\pi'(v|\lambda) = \frac{\partial \pi(v|\lambda)}{\partial \lambda}$ . For the case of a zero inflated geometrical distribution we have  $\pi(v|\lambda) = \lambda(1 - \lambda)^v$  and  $\pi'(v|\lambda) = (1 - \lambda)^{v-1} (1 - \lambda(1 + v))$ . Figure 2 shows the parameter outcome of the minimization procedure applied to experimental data; that is, the vertical offsets resulting from the proposed normal offset transform applied to a large test set of simple geometrical images.

### 3.2 Model for the normal indices $i$

Once we have a model for the vertical offsets, we can deduce a model for the normal indices for a subclass of geometrical images. These images, referred to as Horizon images,  $\mathcal{H}$  exist of two constant colored regions separated by one smooth line discontinuity:

$$\mathcal{H} = \left\{ f_{\Omega}(x, y)_{c(x)} \mid c(x) \in \mathcal{C}^2, f_{\Omega}(x, y)_{c(x)} = \begin{cases} H, c(x) > y \\ 0, c(x) \leq y \end{cases}, H \in \mathbb{R} \right\}. \quad (3.8)$$

For sake of simplicity this section *assumes* a contour running in between the two end points of  $e_{j,k}$ . Remaining scenarios (state) will be discussed at the end of this section. An algorithm to detect in which scenario an edge finds itself in will be given in Section 3.2.1.

The normal indices can be encoded directly with codewords of  $\lceil \log_2 L_{j,k} \rceil$  bits, with  $L_{j,k}$  the number of pixels in the edge  $e_{j,k}$  decaying like  $O(2^{-j})$ . This way of encoding the normal indices presupposes that those indices are distributed uniformly, maximising the entropy. To lower the entropy we derive an appropriate PMF associated with  $i$  for the class of Horizon images which allows an entropy encoder such as an Huffman encoder to lower the expected bitrate. The following lemma connects the PMF to the orientation of an edge somewhere during the decomposition stage assuming the function  $f_e$  above the edge behaves like a step function, which is true for images belonging to  $\mathcal{H}$  and the edge endpoints are located at both sides of the contour. Vertical offsets show up in the derivation since in the discrete setting normal offsets are incapable of exactly pinpointing a function value.

**Lemma 3.1** *Let the image be an element of  $\mathcal{H}$ . Assume  $c(x) = y$  runs in*

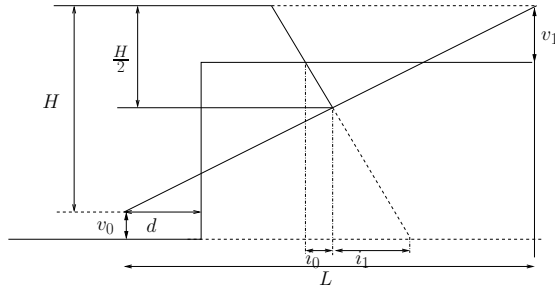


Fig. 3. Situation where piercing points do not take on the exact function value. In the example  $v_0 > 0$  while  $v_1 < 0$ .

between the endpoints of edge  $e$  as depicted in Figure 3. Further assume that the horizontal distance  $d$  between the begin point of the edge and the discontinuity is uniformly distributed on the interval  $[0, l]$ , with  $l$  the euclidian distance in the  $XY$ -plane between the end points of  $e$ :

$$f_D(d) = \frac{1}{l} \chi_A(d) \text{ with } \chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & \text{else} \end{cases}, A = [0, l] \quad (3.9)$$

the signed magnitude of the vertical offsets associated with the begin point and end point of  $e$  are respectively  $v_0, v_1 \in \mathbb{N}$  having a zero inflated geometrical distribution (see Eq. (3.1)) with parameters  $\lambda_0 = \lambda_1 = \lambda$  and  $p_0 = p_1 = p$ . Define  $i_0$  and  $i_1$  as:

$$\begin{aligned} i_0 &= -H/l(H/2 + v_1) \\ i_1 &= H/l(H/2 + v_0) \end{aligned} \quad (3.10)$$

then, for  $0 < i < l/2$ , the PDF is given by:

$$\begin{aligned} f_I(i) &= \frac{\frac{l}{2} - i_1}{l} \left[ (1-p) \frac{\lambda_0}{2} (1 - \lambda_0)^{\left| \frac{l}{H}i - \frac{H}{2} \right|} + p\delta \left( \left| \frac{l}{H}i - \frac{H}{2} \right| \right) \right] \\ &+ \sum_{v_0} f_{V_0}(v_0) \sum_{v_1} \frac{1 - (f_I(i_0(v_1)) + f_I(i_1(v_0)))}{i_1 - i_0} f_{V_1}(v_1). \end{aligned} \quad (3.11)$$

Analogously for  $-\frac{l}{2} < i < 0$ .

**Proof 3.1** Assuming  $v_0, v_1$  independently distributed, the probability mass function  $f_I(i)$  is given by:

$$\begin{aligned}
f_I(i) &= \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} f_I(i|V_0 = v_0, V_1 = v_1) f_{V_0, V_1}(v_0, v_1) dv_0 dv_1 \\
&= \sum_{-\infty}^{\infty} f_{V_0}(v_0) \sum_{-\infty}^{\infty} f_I(i|V_0 = v_0, V_1 = v_1) f_{V_1}(v_1) dv_1 dv_0
\end{aligned} \tag{3.12}$$

with

$$\begin{aligned}
f_I(i|V_0 = v_0, V_1 = v_1) &:= && f_I(i_0(v_1)) \delta_{i_0}(i) \\
&+ && \frac{(1 - [f_I(i_0(v_1)) + f_I(i_1(v_0))])}{i_1 - i_0} \chi_{[i_0, i_1]} \\
&+ && f_I(i_1(v_0)) \delta_{i_1}(i) \\
f_I(i_0(v_1)) &:= && \frac{\frac{l}{2} + i_0(v_1)}{l} \\
f_I(i_1(v_0)) &:= && \frac{\frac{l}{2} - i_1(v_0)}{l}
\end{aligned}$$

The top right picture of Figure 4 shows an instance of the set of distributions given by Eq. (3.12), together with the other possible states an edge can find itself into when it comes into contact with a contour. Lemma 3.2 gives normal index PDF for each of the states. The proof of Lemma 3.2 is similar to the proof of Lemma 3.1.

**Lemma 3.2**

(1) Assume that a discontinuity is present between both edge points then the PDF is given by:

$$a) \quad \text{Equation (3.11)} \quad ; H < l \quad (3.13)$$

$$b) p(i) = \quad 1/l, \quad i \in [-l/2, l/2] \quad ; H \geq l \quad (3.14)$$

$$0, \quad \text{else} \quad (3.15)$$

(2) Assume that one of the endpoints of the edge is located at the discontinuity then:

$$a) p(i) = \frac{1}{2} \left[ (1-p) \frac{\lambda_0}{2} (1 - \lambda_0)^{|\frac{l}{H}i - \frac{H}{2}|} + p \delta \left( \left| \frac{l}{H}i - \frac{H}{2} \right| \right) \right] \quad ; H < l \tag{3.16}$$

$$b) p(i) = \begin{cases} 1/2, & i = \pm l/2 \\ 0, & \text{else} \end{cases} \quad ; H \geq l \tag{3.17}$$

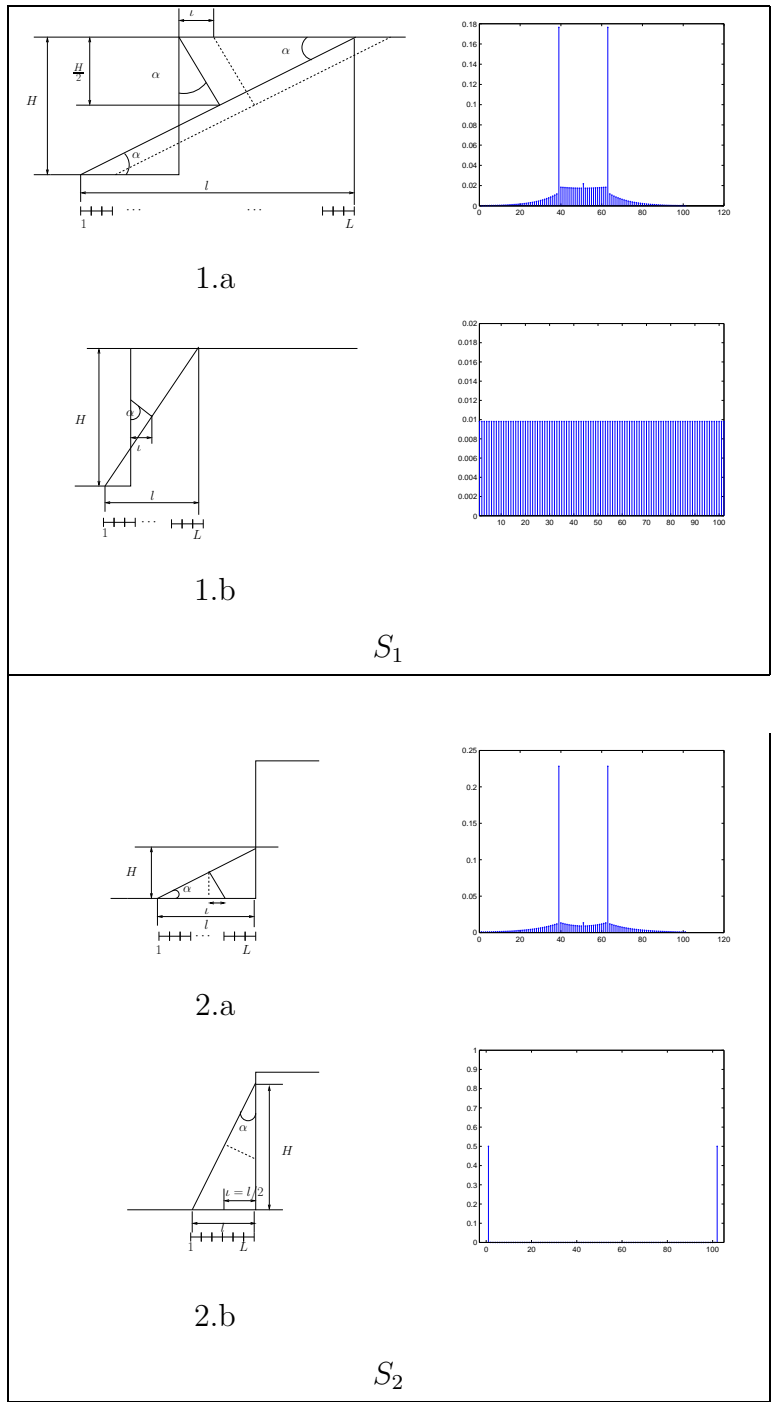


Fig. 4. Four edge vs. step function situations together with their associated PMF for the normal indices given by Eq.(3.14)-(3.17). The two situations at the top of the figure assume a contour runs through the vertical plane containing edge  $e$ . The two situations at the bottom involve an edge where one of the end points is located on the contour. The  $a$ -part of each group concerns an edge where  $H < l$  where the  $b$ -part concerns edges where  $H \geq l$ .

### 3.2.1 Finer scale model selection from given coarse scale situation

Previous section enumerated the different PMF to model the normal indices, given the state of the edge corresponding to the piercing point. We differentiate the cases in three groups. An edge can be *approximately* flat ( $S_0$ ). From Equation (2.4) we see that magnitude of the vertical offset  $v$  is greatly influenced by the slope  $s$  of the normal ray  $r$ . We therefore say that an edge  $e$  is flat ( $\in S_0$ ) if the slope  $s = n_{[x]_{j,k}}/n_{[y]_{j,k}}$  exceeds some value determined by some  $\delta$  such that

$$|s| < \frac{1}{\delta}. \quad (3.18)$$

An edge  $e$  for which Eq. (3.18) holds will be assumed to be flat.

The last two groups fall into the states mentioned in previous section, i.e.  $S_1$  contains both states 1.a and 1.b,  $S_2$  contains states 2.a and 2.b. Once the group of an edge  $e$  is known, the particular instance of the distribution depicted by Figure (4) can be easily determined by  $H_{j,k}$  and  $l_{j,k}$ . The main difficulty is the separation of the 3 groups only at what will be known at the decoder side, i.e.,  $H_{j,k}, l_{j,k}, \dot{l}_{j,k}$ . The flow chart given in Figure 5 gives a rough outline of where the decisions are based upon. The procedure to determine whether or not the

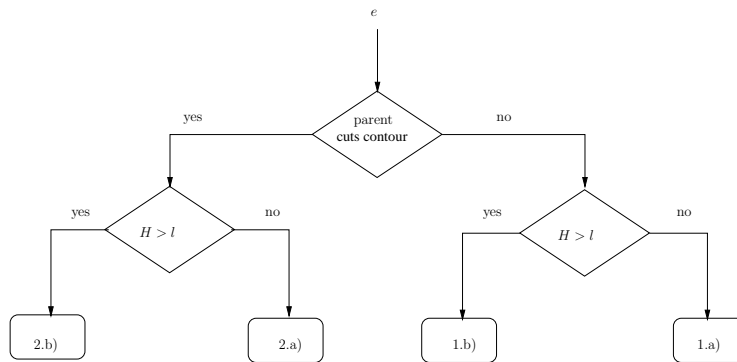


Fig. 5. Flowchart.

edge is crossed by a contour is given by Algorithm (1). Since our coder is a symmetric coder, Algorithm (1) will be traversed in exactly the same way at both the encoder and decoder side. The data needed for a certain decision at the encoder will also be available at the decoder side.

For more natural images there could be more than one contour passing through the vertical plane in between the end points of the edge. In that case it will be possible for an edge to form a merely flat ‘bridge’ spanning over a canyon-like region, although its normal offset can be assigned a large value. These images can be tiled making sure that each tile contains no more than one contour. This can be obtained by a preprocessing procedure, like presented in [17], which constructs the initial coarse mesh adapted to the main features of the function to be processed.

---

**Algorithm 1** Group selection procedure

---

```
procedure PARENT_STATUS( $e_{j,k}$ )  
  if  $e$  has no parents then  
    goto: NO_PARENT( $e_{j,k}$ )  
  else if The normal ray of the parent of edge  $e_{j,k}$  has not pierced a contour  
  then  
    goto: PARENT_DID_NOT_PIERCED_CONTOUR( $e_{j,k}$ )  
  else  
    goto: PARENT_PIERCED_CONTOUR( $e_{j,k}$ )  
  end if  
end procedure
```

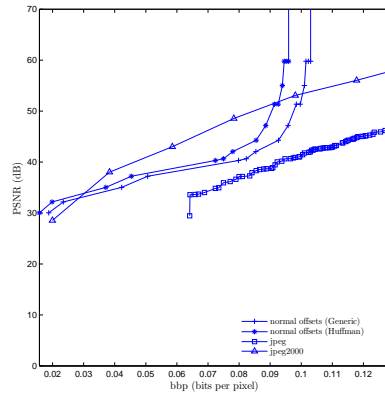
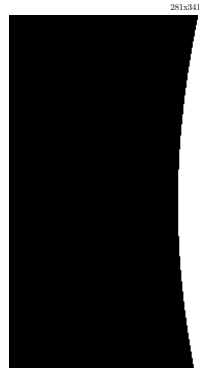
```
procedure NO_PARENT( $e_{j,k}$ )  
  goto:  $S_0\_OR\_S_1$ ( $e_{j,k}$ )  
end procedure
```

```
procedure PARENT_DID_NOT_PIERCED_CONTOUR( $e_{j,k}$ )  
  if the parent edge belongs to  $S_0$  then  
    goto:  $S_0\_OR\_S_1$ ( $e_{j,k}$ )  
  else if  $H_{j-1} < l_{j-1}$  then  
    if  $i < H_{j-1}^2 / (2l_{j-1}) \times L_{j-1} / l_{j-1}$  then  
      goto:  $S_0\_OR\_S_2$ ( $e_{j,k}$ )  
    else  
      goto:  $S_0\_OR\_S_1$ ( $e_{j,k}$ )  
    end if  
  else  
     $e \in S_2$   
  end if  
end procedure
```

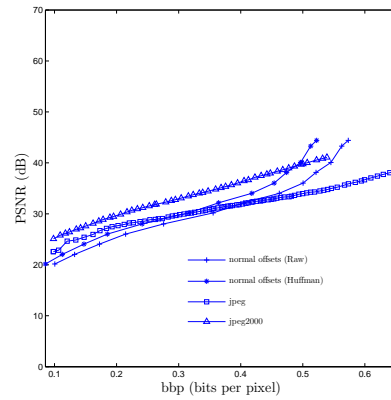
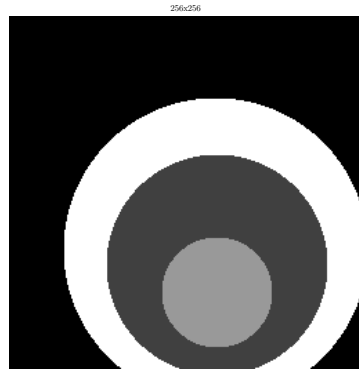
```
procedure PARENT_PIERCED_CONTOUR( $e_{j,k}$ )  
  goto:  $S_0\_OR\_S_2$ ( $e_{j,k}$ )  
end procedure
```

```
procedure  $S_0\_OR\_S_1$ ( $e_{j,k}$ )  
  if slope of  $r$  satisfies Equation (3.18) then  
     $e_{j,k} \in S_0$   
  else  
     $e_{j,k} \in S_1$   
  end if  
end procedure
```

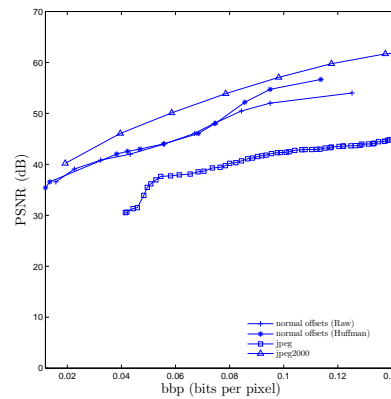
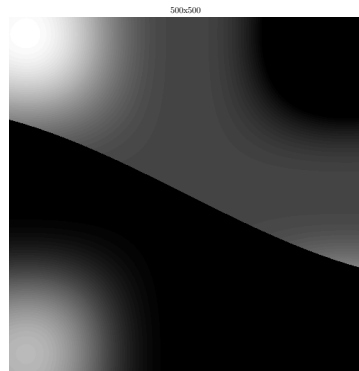
```
procedure  $S_0\_OR\_S_2$ ( $e_{j,k}$ )  
  if slope of  $r$  satisfies Equation (3.18) then  
     $e_{j,k} \in S_0$   
  else  
     $e_{j,k} \in S_2$   
  end if  
end procedure
```



'Horizon Image'



'Circles Image'



'Piecewise Smooth'

Fig. 6. The rate distortion curve of a JPEG encoder and a JPEG2000 encoder together with the rate distortion curve of a normal offset encoder without using offset modelling (Raw) and a normal offset encoder using the proposed models as input for an Huffman entropy encoder (Huffman). Note that the gain entropy coding of the non Horizon images kicks in at higher bit rates, when the fine representation of the contour becomes more and more important.



## 4 Results and Further research

Recently much research has been done to solve the problem of sparsely representing two dimensional functions having line discontinuities. Since many images consist of large smooth areas separated by smooth contours, they can be interpreted as a regular sampling of such functions. Current transform coders based on wavelet decompositions have a suboptimal  $n$ -terms approximation rate for such images. They fail to produce sparse representations of smooth line discontinuities.

We proposed a nonlinear refinement procedure based on normal mesh techniques to sparsely represent contours. We adapted the nonlinear transform towards the digital setting.

We proposed a model for the wavelet coefficients coming from our encoder and added an entropy encoder to reduce the bit-rate. Here we show several results on the performance of the proposed encoder.

For the experiments depicted in Figure 6 we used the  $L_2$ -norm (MSE). Using the  $L_1$ -norm, for which the transform performs best, the outcome of our experiments would be more favorable, since in contrast to  $L_1$ , the  $L_2$ -norm prefers a lot of little errors over a few big errors. The experiments shown in Figure 6 indicate that the current normal offset encoder as described here does not yet outperform the JPEG2000 encoder over the entire range. At low rates the strong aliasing effect of the scheme being a refinement scheme plays the dominant role. The justification for this is that for the entire range, so even for low resolution triangulations, the vertical offsets are stored losslessly while they are not the main contributors of the image quality. Once the mesh is fine enough the distortion decays rapidly. Including a rate distortion optimised quantiser ([16]) on the vertical offsets starting from the lossless encoded image will improve the performance at lower rates.

Around the lossless rate a strong convergence is seen. For simple geometrical images, the rate at which a lossless compression is achieved is much lower than the JPEG2000 as shown in Table 1.

Image	JPG	JPEG2000	Raw	Huffman
horizon	2.42	1.70	1.47	<b>1.33</b>
circles	9.97	7.28	4.58	<b>4.17</b>
cameraman	<b>22.5</b>	32	93	75.8

Table 1

File size in kilo bytes of lossless encoded test images (Figure 6) using different encoders. We note that for ‘simple’ geometrical images the normal offset based algorithms outperform both jpeg standards. For more complex images the jpeg standards still perform better, since our encoder has not yet incorporated a segmentation procedure to create an appropriate initial mesh where the rest of the refinement procedure is based upon.

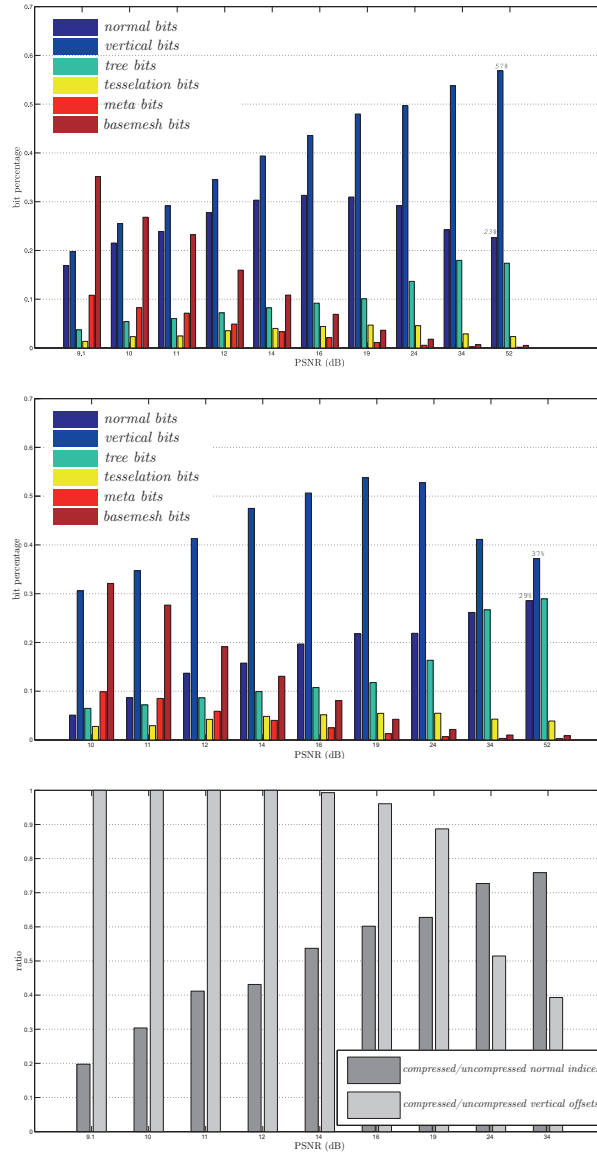


Fig. 7. Percentage of the bits allocated to different coefficients for the bottom (‘circles’) image of Figure 6. (Top) The vertical offsets where *not* entropy coded. The normal offsets where coded with a number of bits decaying like  $\log_2(\mathcal{O}(2^{-j}))$ . (Bottom) The bottom Figure shows the reduction in both vertical and normal offsets when entropy coding is used as explained in Section 3.

Figure 7 illustrates the gain in the relative bit budget consumed by both normal and vertical offsets using the proposed model. Further research has to be done to finetune different parameters on the level of the transform as well as on the bitcoding level. For instance more suitable and image specific basemeshes can be used instead of the trivial diagonal split of the domain. Parameters  $p, \lambda$  described in Section 3 can be adjusted for each image and could be made ‘resolution level dependent’. The ‘ad hoc’ algorithm described in Section 3.2.1 to select the appropriate probability model to encode a certain

offset has to be made more robust. Those measures can still further improve the performance of the proposed encoder.

## 5 Acknowledgments

The work is supported by the Fund for Scientific Research (FWO) project SMID: Stability of Multiscale Transforms on Irregular Data, grant #G.0431.05 and the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian Federal Science Policy Office and by the Center Of Excellence on Optimisation in Engineering of the K.U.Leuven.

## References

- [1] D.K. Agarwal, A.E. Gelfand, and S. Citron-Pousty. Zero-inflated models with application to spatial count data. *Environmental and Ecological Statistics*, 9(4):pp. 341–355, December 2002.
- [2] P. Schröder, A. Khodakovskiy, and Wim Sweldens. Progressive geometry compression. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 271–278. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.
- [3] 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.
- [4] I. Daubechies, O. Runborg, and W. Sweldens. Normal polyline approximation. *Constructive Approximation*, 20(3):399–463, May 2004.
- [5] S. Dekel and D. Leviatan. Adaptive multivariate approximation using binary space partitions and geometric wavelets. *SIAM J. Numer. Anal.*, 43(2):707–732, 2005.
- [6] R. A. DeVore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998.
- [7] M. N. Do and M. Vetterli. Contourlets. In G. Welland, editor, *Beyond Wavelets*, pages 83–107. Academic Press, 2003.
- [8] M. Antonini, F. Payan. An efficient bit allocation for compressing normal meshes with an error-driven quantization. *Computer Aided Geometric Design*, 22:466–486, 2005.
- [9] Ilja Friedel, Peter Schröder, and Andrei Khodakovskiy. Variational normal meshes. *ACM Trans. Graph.*, 23(4):1061–1073, 2004.

- [10] I. Guskov, K. Vidimčė, W. Sweldens, and P. Schröder. Normal meshes. In *SIGGRAPH 2000 Conference Proceedings*, 2000.
- [11] M. Jansen, R. Baraniuk, and S. Lavu. Multiscale approximation of piecewise smooth two-dimensional functions using normal triangulated meshes. *Appl. Comp. Harm. Anal.*, 2005.
- [12] A. Khodakovsky and I. Guskov. Compression of normal meshes. In *Geometric Modeling for Scientific Visualization*, pages 189–207. Springer Verlag, 2003.
- [13] E. Le Pennec and S. Mallat. Sparse geometric image representation with bandelets. *IEEE Trans. on Image Processing*, 14(4):423–438, 2005.
- [14] H. Choi S. Lavu and R. Baraniuk. Geometry compression of normal meshes using rate-distortion algorithms. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 52–61, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [15] E. Simoncelli. Modeling the joint statistics of images in the wavelet domain. In M. A. Unser, A. Aldroubi, and Laine A. F., editors, *Wavelet Applications in Signal and Image Processing VII*, volume 3813 of *SPIE Proceedings*, pages 206–214, July 1999.
- [16] Gary J. Sullivan and Shijun Sun. On dead-zone plus uniform threshold scalar quantization. In *Visual Communications and Image Processing 2005, Proc. of SPIE*, volume 5960.
- [17] W. Van Aerschot, Vanreas E., and Bultheel A. Preprocessing for the decomposition of images with normal offsets. Technical Report TW 475, Department of Computer Science, K.U.Leuven, 2006.
- [18] W. Van Aerschot, M. Jansen, E. Vanraes, and A. Bultheel. Image compression using normal mesh techniques. In A. Cohen and L. Schumaker, editors, *Curves and Surfaces, Avignon 2006*, Brentwood, 2006. Nashboro Press. Accepted.