Surface Compression With Geometric Bandelets

Gabriel Peyré, Stéphane Mallat

Surface Compression with Geometric Bandelets

Gabriel Peyré* Stéphane Mallat
CMAP, École Polytechnique

Abstract

This paper describes the construction of second generation bandelet bases and their application to 3D geometry compression. This new coding scheme is orthogonal and the corresponding basis functions are regular. In our method, surfaces are decomposed in a bandelet basis with a fast bandeletization algorithm that removes the geometric redundancy of orthogonal wavelet coefficients. The resulting transform coding scheme has an error decay that is asymptotically optimal for geometrically regular surfaces. We then use these bandelet bases to perform geometry image and normal map compression. Numerical tests show that for complex surfaces bandelets bring an improvement of 1.5dB to 2dB over state of the art compression schemes.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—surface and object representations

Keywords: Bandelets, discrete multiscale geometry, geometry image, normal map, compression.

1 Discrete Multiscale Geometry

The geometry of natural surfaces is complex and intrinsically multiscale [Dana et al. 1999]. The different techniques used to describe this geometry must account for a variety of structures at different levels of detail: macro structures (the traditional 3D mesh representation), mesostructures (bump map or displacement map), and micro-scale material structures (reflectance function). In this paper we will focus on the problem of *compression* of these geometric structures and provide a new representation called *second generation bandelets*.

Geometry Drives Creation and Computation The geometric features of a surface are at the heart of the design process, both for artists and for CAD editing: human perception is mainly sensitive to lines of curvature which create most lighting effects. In the processing steps, most algorithms take special care of these geometric features, both for efficiency (anisotropy in meshing [Alliez et al. 2003]) and aesthetic considerations (non-photorealistic rendering [Hertzmann and Zorin 2000]).

Geometry is Discrete Working with digital data means working in a discrete setting. Even if the underlying functional model is continuous, the first step before any processing of the surface is a sampling stage which can model various acquisition processes (range scanning [Levoy et al. 2000], reconstruction from photographs [Slabaugh et al. 2001], etc) and remeshing of the 3D model (see e.g. [Alliez et al. 2003]). A wavelet transform can then be applied on the discrete samples to obtain a multiresolution representation of the original data [Alliez and Gotsman 2005].

Geometry is Multiscale The geometry of surfaces is naturally multiscale, as one can see it directly on various surfaces. On finely scanned models (acquired for artistic studies [Levoy et al. 2000] or for reverse engineering [Hoppe et al. 1994]), sharp features are often located on a width of a few vertices. For such surfaces, the geometry is not a collection of discontinuities, but rather areas of high curvature. Locating these geometric features is an ill-posed problem [Ohtake et al. 2004], and we will explain in this paper how to recast it into a *regularity direction* estimation, which is well-posed and optimal for the purpose of surface compression.

Image-based Surface Processing The use of regular grids for surface representation and compression is a unifying framework for most 3D surface data. We will use it to propose a common functional model for geometric surfaces, formulate the coding problem, and provide an algorithmic tool to solve the problem. By fixing the parameterization of the 3D model we are able to use classical techniques from harmonic analysis such as orthogonal projections and best basis algorithms. Although the final distortion we want to minimize is geometric (e.g. Hausdorff distance), we will target the L^2 norm for stating the main theorem, and exploit the degree of freedom in bit allocation to optimize the geometric error.

The main image-based representations for surfaces include:

- **Geometry images** [Gu et al. 2002]: the 3D geometry is resampled on a regular grid, and is compactly encoded as an RGB image. It is mostly used to represent large scale features of the mesh (overall shape and deep creases).
- **Normal map** [Peercy et al. 1997]: to model fine scale details of the geometry, the normals are encoded with a high sampling rate in an RGB image. On recent hardware, per-pixel bump-mapping can be done in real time with normal maps.
- **Other maps**: one can cite texture mapping, displacement map [Wang et al. 2003], environnement mapping [Agarwal et al. 2003], volumetric textures [Owada et al. 2004], etc.

In this paper, we will focus on the compression of geometry images and normal maps. However, as soon as some geometric regularity exists in a map (e.g. creases in displacement map) our bandelet scheme will provide similar enhancement as well.

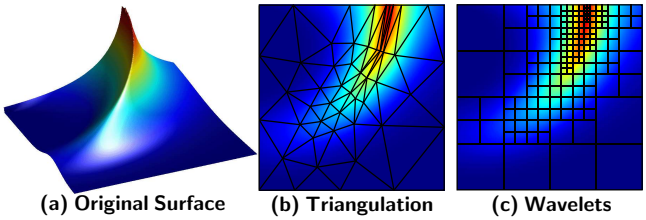
2 Geometric Compression of Surfaces

Surface Functional Model In this paper, we model a function $f : [0,1]^2 \mapsto \mathbb{R}$ with geometric regularity as C^2-regular outside a set of edges which are themselves C^2-regular curves. In the continuous setting, we consider a function from $[0,1]^2$ to $\mathbb{R}^3$ which can model either a geometry image or a normal map. We treat each component of this 3D-valued function independently, and assume each has a geometric regularity.

However, natural surfaces often do not have sharp discontinuities, so the model also includes some smoothing by an unknown kernel. The resulting functions can be written as a convolution

*e-mail:gabriel.peyre@polytechnique.fr

$f = \widetilde{f} * h$ where $\widetilde{f}$ is a function with sharp features (regular outside a set of edges) and h is the unknown smoothing kernel. We will call this class of functions C^2*-geometrically regular functions*. Figure 1, (a), shows such a function that has a sharp feature that smoothly vanishes (the size of the kernel is progressively increasing along the edge).



(a) Original Surface (b) Triangulation (c) Wavelets

Figure 1: *Approximation of a surface with a fading-away edge. Only the support of the underlying basis functions are depicted.*

In graphics, uniformly regular surfaces were the first functional model proposed for 3D surfaces, such as NURBS in CAD modeling [Farin 1993]. Later, the introduction of subdivision surfaces for modeling [Biermann et al. 2000] and reconstruction [Hoppe et al. 1994] enabled the description of complex piecewise smooth surfaces. The geometrically regular functional model we propose in this paper allows us to replace sharp discontinuities with semi-sharp creases. This model has already been used successfully for character modeling and animation [DeRose et al. 1998].

Classical Isotropic Surface Compression Many methods have been proposed to solve the problem of 3D geometry compression, see the recent survey of [Alliez and Gotsman 2005]. In order to perform such compression, one often relies on a semi-regular remeshing of the original model. The first construction of wavelets on triangulations was built by encoding the differences with respect to a subdivision scheme [Eck et al. 1995]. The lifting scheme [Schröder and Sweldens 1995] extends this construction and provides a useful tool for surface analysis. In fact, best known coders include the normal multiresolution construction of [Khodakovsky and Guskov 2003]. All these schemes are closely related to classical wavelet bases on regular grids, and few theoretical results exist for these surface-based constructions (except for the 2D case, see [Daubechies et al. 2004]).

For images and geometry images compression, currently the most efficient algorithms are transform codes that decompose the signal in an orthonormal (or nearly orthonormal) basis and quantize the resulting coefficients. JPEG and JPEG2000 are examples of such algorithms. Using regular basis functions is also necessary to avoid introducing blocking artifacts in the compressed signal.

When one considers the class of C^2-geometrically regular surfaces with sharp or semi-sharp features, represented with a geometry image f and a normal map g, the best wavelet transform codes f_R and g_R with R bits satisfies

$$\|f - f_R\|^2 \leqslant CR^{-3/2}\log^{3/2}(R) \quad \text{and} \quad \|g - g_R\|^2 \leqslant CR^{-1}\log(R),$$

where C is a constant that depends only on f. This difference is due to the fact that geometry images only have discontinuities of tangents whereas normal maps have step discontinuities. These approximation rates are not optimal. In order to exploit the regularity that exists along sharp or semi-sharp transitions, one has to build elongated functions that are adapted to the geometry of the surface. In this paper we construct a new class of orthogonal bases that exhibits the optimal $R^{-2}\log^2(R)$ transform coding rates for geometry images and normal maps.

Surface Approximation and Anisotropy Geometric approximation problems have been well studied, and although the construction of an optimal triangulation is NP-hard [Agarwal and Suri 1998], some effective greedy solutions exist, such as [Hoppe 1996; Garland and Heckbert 1997; Lindstrom and Turk 1998], and more global error-driven methods [Cohen-Steiner et al. 2004] are promising. The problem is even more complicated for semi-sharp features, because the aspect ratio of each triangle has to be tuned adaptively depending on the unknown smoothing kernel h (see figure 1, (b)).

In order to have a well-posed problem (with polynomial complexity), one must fix a scale for the geometry and build an approximation for this scale. In computer graphics, the problem of sharp features detection is well studied [Ohtake et al. 2004], and involves some smoothing to fix the scale of the salient structures. For the bandelet scheme, with coding efficiency in mind, we will follow this paradigm and show that working at a fixed scale 2^j (constructing a different geometry for each scale) is equivalent to regularizing the geometry (allowing error in location up to 2^j), which in turn makes the optimization tractable in nearly linear time complexity.

In the image analysis community, some recent constructions are able to capture this geometric regularity. Most notably are the curvelets [Candès and Donoho 1999], contourlets [Do and Vetterli 2005], wedgeprints [Wakin et al. 2005] and non-linear subdivision schemes [Matei and Cohen 2002]. However, none of these schemes is able to construct orthogonal bases of regular functions which is highly desirable for compression.

3 The Bandelet Approach

The bandelet approximation scheme, introduced in [Le Pennec and Mallat 2004], takes advantage of geometric image regularity by removing the redundancy of a warped wavelet transform by performing a *bandeletization*. Unfortunately, the resulting transform is non orthogonal and the warping introduces boundary artifacts. Instead, our second generation bandelet transform is constructed over a standard orthogonal wavelet transform. It is thus simpler, orthogonal, and without border effect. We implement this second generation bandeletization first by reordering the 2D wavelet coefficients and then performing a 1D wavelet transform.

Wavelet Algorithms Both 1D and 2D wavelet transforms are explained in [Mallat 1998], and we will just recall some basic facts in the 2D setting.

The wavelet transform of a function $f : \mathbb{R}^2 \mapsto \mathbb{R}$ is the decomposition of f on an orthogonal basis composed of translates and dilates of three mother wavelets $\{\psi^H, \psi^V, \psi^D\}$ (for the horizontal, vertical and diagonal directions). More formally, it is the set of dot products

$$\langle f, \psi^s_{jn}\rangle \quad \text{with} \quad \begin{cases} s \in \{H,V,D\},\ j \in \mathbb{Z},\ n \in \mathbb{Z}^2, \\ \psi^s_{jn}(x) = \frac{1}{2^j}\psi^s(2^{-j}x_1 - n_1, 2^{-j}x_2 - n_2). \end{cases}$$

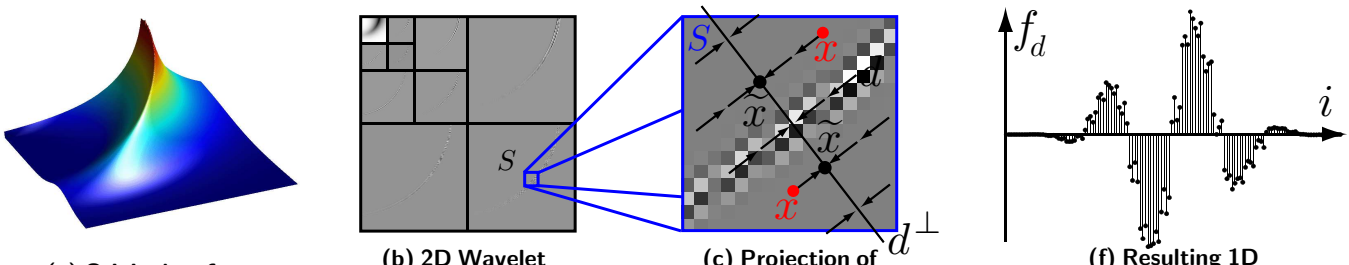
The function ψ^s_{jn} is supported near the point $2^j n$ on a square of width $\sim 2^j$. In the discrete setting, the wavelet transform takes an image of N pixels and computes the set of N dot products

$$\begin{cases} \langle f, \psi^s_{jn}\rangle \ \text{ for } \ 2^{-J} \leqslant 2^{-j} < \sqrt{N}, \ \text{ and } \ 0 \leqslant n_1, n_2 < 2^{-j}, \\ \langle f, \varphi_{Jn}\rangle \ \text{ for } \ 0 \leqslant n_1, n_2 < 2^{-J}, \end{cases}$$

where the projection on φ_{Jn} functions produces a coarse approximation at scale 2^J. This scale 2^J represents the level at which we stop the wavelet transform (one could go all the way down to $J = 0$). These values can be conveniently stored in an array of N pixels as shown on figure 2 (b). Note that the square in the upper left corner of the transformed image contains the dot products with the functions $\{\phi_{Jn}\}_n$ (coarse scale approximation).

Transform coding in a wavelet basis performs an adaptive approximation using functions with square supports of various sizes, as shown in figure 1 (c).

2D Wavelet Transform Working in the wavelet domain allows us to treat each scale of the transform independently. For our dis-



(a) Original surface (b) 2D Wavelet transform (c) Projection of sampling points (f) Resulting 1D signal

Figure 2: *Discrete reordering of the sampling points.*

cussion, we will use a fixed scale 2^j, which is equivalent to selecting the three sub-images in the wavelet transform containing the coefficients $\langle f, \psi^s_{jn} \rangle$ for $s \in \{V,H,D\}$. Also, we denote f a C^2-geometrically regular function as previously defined.

A wavelet transform is able to compress the regular parts of the surface well. On figure 2 (b), one can see that the only non-zero wavelet coefficients are close to the singularity. To achieve a better compression of the original surface, we focus on these high coefficients at scale 2^j and re-transform them to obtain a great number of values that are close to zero.

This bandeletization removes the correlation that exists between wavelet coefficients near the singularity. In order to remove this redundancy, we must find some regularity in the wavelet transform of the surface and construct an adapted approximation scheme. There are in fact two sources for this regularity that we can use:

- **Regularity due to the wavelet**: dot products with a set of translated functions can be computed using a *convolution*:

$$\langle f, \psi^s_{jn} \rangle = f * \psi^s_j(n_1 2^j, n_2 2^j) \quad \text{where} \quad \psi^s_j(x) = \tfrac{1}{2^j}\psi^s(-2^{-j}x),$$

  and $s \in \{V,H,D\}$. As a result, although the original function can be singular at edge locations, the wavelet coefficients are samples of a regularized function f^s_j obtained by convolving the original function f with the "blurring" kernel ψ^s_j of width $\sim 2^j$. This blurring ensures a regularity in the direction orthogonal to the geometry. This is important because it allows us to make some errors in the localization of the exact geometry without too much impact on the approximation.
- **Regularity along the geometry**: the function f is regular in the direction of the geometry. Indeed, when one moves parallel to the geometric curve, the transformed function $f * \psi^s_j$ is smoothly varying. We use this regularity to compress the remaining non-zero wavelet coefficients.

These two sources of regularity work hand in hand to make the compression algorithm fast. Indeed, the first regularity allows us to make small mistakes which enables us to extract the geometric regularity quickly, but still precisely.

Reordering of the Grid Points (bandeletization, step 1) Until the end of the section, we select a square S of width L in the wavelet domain at some scale 2^j and orientation $s \in \{H,V,D\}$. Our goal is to clear the anisotropic redundancy that has not been removed by the 2D wavelet transform. We want to find a correct numbering of the grid points so that the 1D discrete signal obtained from this reordering is smooth.

This reordering must be described in a simple manner, more formally it should be parameterized by a small number of parameters. To recover the geometric regularity that exists around sharp features, our scheme is based on directional projections. The reordering will be described by a single direction d, that should be as parallel as possible to the real geometry. As shown on figure 2 (c), we select each sampling location x of the regular grid, and project it orthogonally onto the line $d^\perp$ to get a new point $\tilde{x}$. To construct a discrete 1D signal we now order the points $\tilde{x}$ according to their numbering along the axis $d^\perp$. The point $\tilde{x_i}$ is thus the ith point along $d^\perp$. We get a new 1D discrete signal f_d defined by

$$\forall i, \quad f_d[i] = f(x_i).$$

Note that the only thing that matters is the relative position of the sampling locations, and the resulting signal is considered as if it was regularly sampled.

1D Wavelet Transform (bandeletization, step 2) We use a 1D wavelet transform to compress the 1D discrete signal f_d, see [Mallat 1998]. This transform also provides an effective way to discriminate between good and bad reorderings. To that end, the user provides a threshold T that is the only input parameter of our bandelet algorithm and controls its compression rate. Bandelet coefficients bellow T will be discarded, so that higher T gives rise to a more agressive compression.

For each square S and direction d, we are able to compute a reordering that shuffles the points inside S and produces a 1D signal f_d. We are then left with two questions:

- What should be the size of the square S?
- What should be the direction d in each square?

Hopefully, the 1D wavelet transform provides a simple way to discriminate between the possible S and d. Our goal is to find a reordering, or equivalently a size of square S and a direction d, that produces as few as possible 1D wavelet coefficients greater than T.

Figure 3 shows how this 1D wavelet transform is able to choose both an admissible orientation and the right size of the square. On top row one can see various squares S extracted around a singularity in wavelet space. On middle row the discrete signal f_d is shown (only the central part is depicted). On bottom row on can see the magnitude of the 1D wavelet coefficients of f_d.

(a) The square is too small, since extending a bit S does not produce additional wavelet coefficients b_k such that $|b_k| > T$.

(b) The square is too big, there is too much coefficients b_k above T. This is because the real geometry has some curvature and the direction d deviates too much from it.

(c) The square has the correct size, and the 1D signal f_d is smooth. Note that there are much more b_k satisfying $|b_k| < T$ than original coefficients $f_d[i]$ with $|f_d[i]| < T$.

(d) The direction d deviates too much from the real geometry.

Note that this choice of S and d is relative to the precision T, and squares that are correct for a large T (agressive compression) can be too big for a smaller T since more precision is needed.

The best d is the once that leads to the best compressed representation after the bandeletization. If there is no preferential orientation in the square S then it is better not to perform any bandeletization. In this case the bandelet transform is a standard 2D wavelet transform, and we set $d = \text{NULL}$.

We are still stuck with the problem of designing the exact partition into squares of the wavelet domain at scale 2^j. What we are looking for is the partition that gives rise to the best compression of the surface. We will see in sections 4 and 5 how our algorithm can perform this construction in a provably optimal manner.

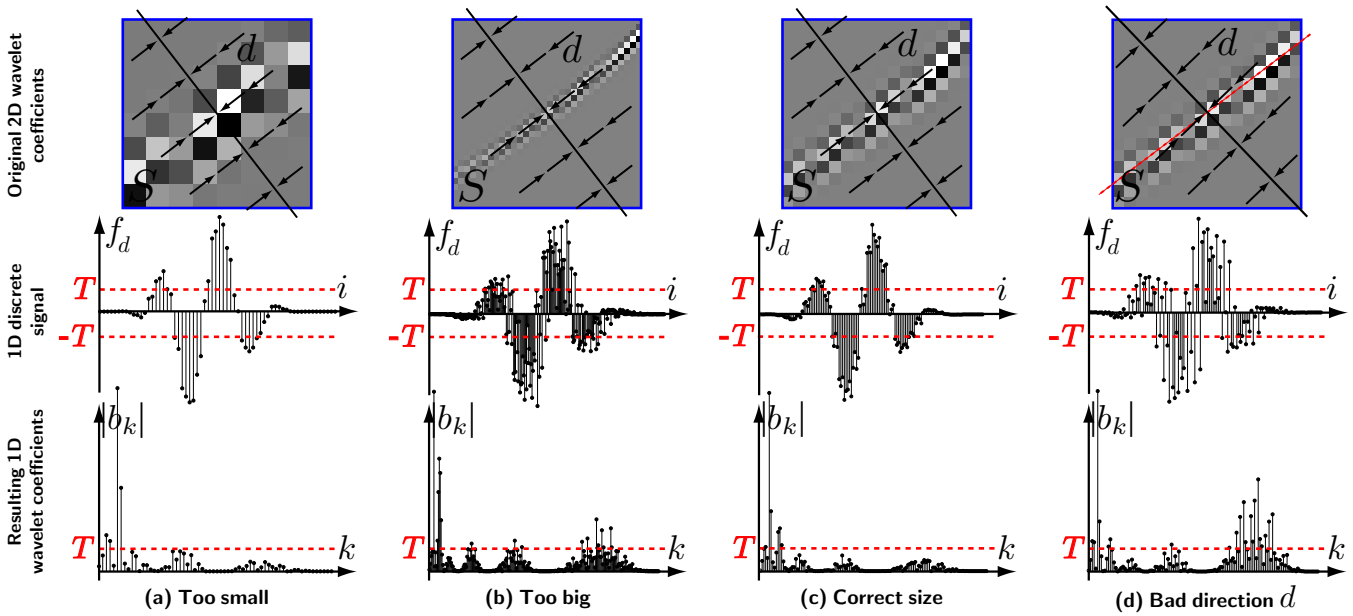


Figure 3: *Influence of the size of S and of the direction d.*

4 Bandelet Approximation Algorithm

We will now explain in details the different steps of the bandelet approximation algorithm, see also figure 4. A Matlab implementation of this algorithm is freely available [Peyré and Mallat 2005a].

(1) Input of the algorithm. The user provides a surface (or normal map), stored as an 3-channels image using any geometry image method. He also provides a threshold T that controls the compression rate of the algorithm.

(2) 2D wavelet transform. We first compute the 2D wavelet transform of the original image f. This transform can be either orthogonal or biorthogonal, and is applied to each of the 3 channels of the image. This results into a collection of 3-tupple of images (f_j^H, f_j^V, f_j^D). The new images f_j^s, for each scale 2^j and orientation $s \in \{V,H,D\}$, are stored in a single image of the same size as the original image f, see also figure 2 (b). The following steps (3)-(7) implement the bandeletization, which is repeated for each scale and orientation.

(3) Selecting each dyadic square. A dyadic square is by definition a square obtained by recursively splitting the original wavelet transformed image f_j^s into four sub-squares of equal size. We restrict ourselves to squares of width L pixels with $4 \leqslant L \leqslant 2^{-j/2}$. The following steps (4)-(7) are repeated for each dyadic square S at a given scale 2^j and orientation s of the wavelet transform.

(4) Selecting each geometry. There are as many possible 1D reordering of the grid points as directions d joining pairs of points in the square S of width L. The number of potential directions is less than $2L^2$. The following steps (5)-(7) are repeated for each of these potential directions d.

(5) Projection of the sampling locations. We now perform the 1D discrete reordering of the sampling location explained in the previous section. This is done by projecting the sampling location along d and sorting the resulting 1D points from left to right.

(6) 1D resulting signal. This 1D numbering of the sampling points defines a 1D discrete signal f_d as explained in the previous section

(7) 1D wavelet transform. We perform a 1D discrete wavelet transform of f_d.

(8) Selection of the best geometry (not shown). For a given threshold T, we have to choose the direction d which generates the less approximation error. In the following, we denote by $\{b_k\}$ the coefficients of the 1D wavelet transform of f_d, and by R_B the number of bits needed to code the quantized coefficients $\{Q_T(b_k)\}$. We use a nearly uniform quantizer

$$\begin{cases} Q_T(x) = 0 & \text{if} \quad |x| \leqslant T, \\ Q_T(x) = \text{sign}(x)(q+1/2)T & \text{if} \quad qT \leqslant |x| < (q+1)T. \end{cases}$$

To select the best geometry, we must choose the direction d that minimizes the Lagrangian

$$\mathscr{L}(f_d, R) = \|f_d - f_{dR}\|^2 + \lambda T^2 (R_G + R_B),$$

where f_{dR} is the signal recovered from the quantized coefficients $\{Q_T(b_k)\}$ using the inverse 1D wavelet transform, and R_G is the number of bits needed to code the geometric parameter d with an entropy coder. We use $\lambda = 3/28$, see [Le Pennec and Mallat 2004] for a justification of this value.

(9) Output of the transform. The resulting 1D wavelet coefficients $\{b_k\}$ corresponding to the best geometry d can be stored in a 2D image of the same size as S. We use a zig-zag scanning order, so that low-scale wavelet coefficients b_k are stored in the upper-left corner of the output square. The fact that high coefficients usually correspond to these scales can be exploited in the arithmetic coder that codes bandelet coefficients. When comparing the images in steps (3) and (9), we can notice that the anisotropic redundancy has almost disappeared.

(10) Build the quadtree (not shown). Once we have computed the approximation over each dyadic square, we must choose the best layout of squares. This is explained in the next section.

What Do Bandelets Look like? A second generation bandelet transform is a 2D wavelet transform followed by a bandeletization. The transformation of a function f using this algorithm is equivalent to a decomposition of f on a bandelet basis $\mathscr{B}$. The bandelet functions b_μ are specified by $\mu = (j, S, k, m)$ where

- 2^j is the scale of the 2D wavelet transform,
- S is a dyadic square of width L pixels, with $1 \leqslant L \leqslant 2^{-j}$,
- $k \in \{0, \dots, 2\log_2(L)\}$ and $m \in \{1, \dots, 2^k\}$ are the scale and index in the 1D wavelet transform.

The continuous underlying bandelet functions are roughly contained in a band of width $2^j L$ and of height 2^{j+k}, but the functions

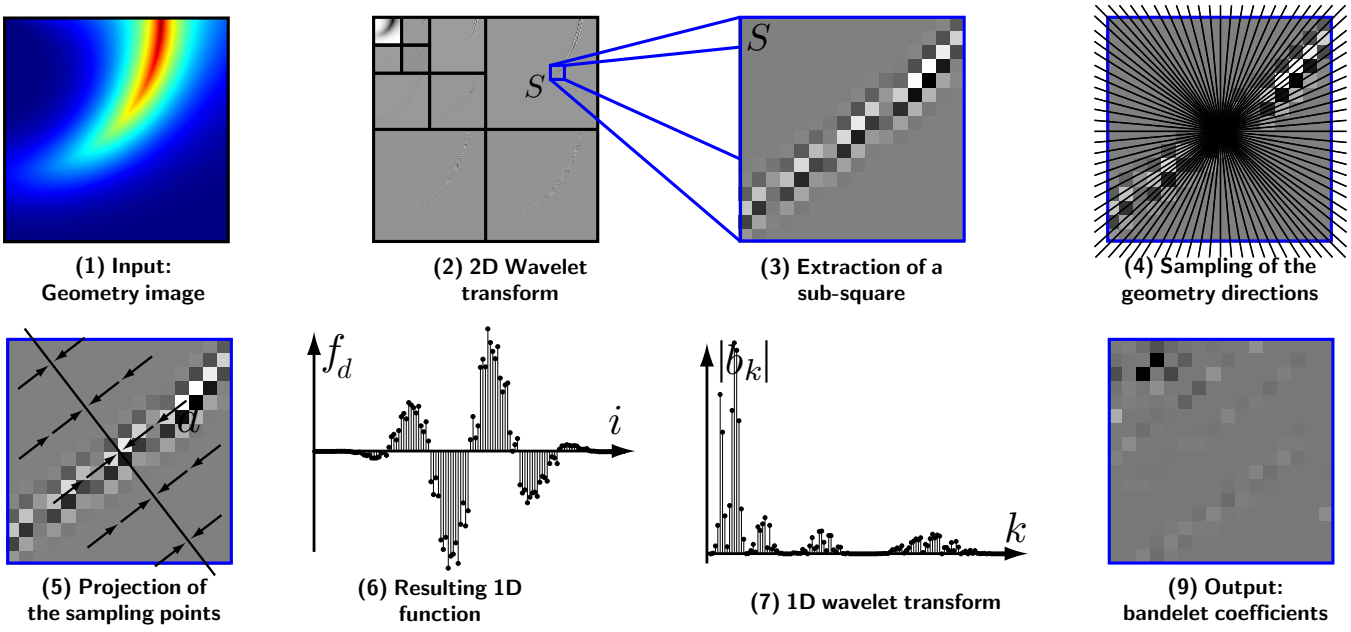


Figure 4: *Overview of the algorithm.*

overlap each other. This is because the bandelet transform can be written as the succession of two transforms (2D wavelet and 1D wavelet along the geometry). This leads to several important remarks:

- The bandelet functions are as *regular* as the underlying wavelet functions.
- Although each quadtree segments the space in non-overlapping squares, the bandelet reconstruction does not suffer from blocking artifact. This is because the block-reconstruction in wavelet space is filtered through the wavelet transform.
- Working at fixed scale regularizes the geometry and the corresponding reconstruction

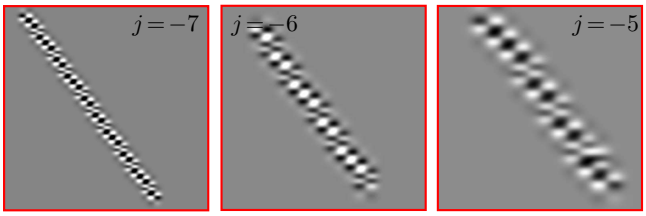


Figure 5: *Graphical display of continuous bandelet functions for various scales* 2^j.

5 Construction of the Quadtree

In the previous section, we have presented the first part of the bandelet algorithm that computes a bandelet transform over each dyadic square at each scale 2^j in the wavelet domain. This is of course a redundant transform, and we must choose a layout of squares that forms the best segmentation of each scale. Such a segmentation is conveniently represented as a quadtree. The second part of the transform builds the best quadtree in a provably optimal manner, using a Lagrangian optimization on the quantized geometry and bandelet coefficients.

Specification of a Bandelet Basis Figure 6 (bottom) shows two different quadtrees for the finest scale 2^j. Once a quadtree is chosen, the complete bandelet transform discards the transformed coefficients, produced by step (8), that do not belong to squares S in the quadtree. This bandelet transform is an orthogonal (resp. biorthogonal) transform since the 1D and 2D wavelet transforms are orthogonal (resp. biorthogonal).

Once we have chosen a segmentation for each scale 2^j and an approximate geometry direction d inside each square, we have the associated bandelet basis $\mathscr{B} = \{b_\mu\}_\mu$, where μ is some parameter indexing the vectors of the basis. The bandelet transform computes the projection of a function f on this basis, i.e. the set of dot products $\{\langle f, b_\mu\rangle\}_\mu$, using the algorithm described in the previous section. A complete bandelet representation is thus composed of:

- A quadtree segmentation for each scale 2^j.
- For each scale 2^j and each dyadic square in the quadtree:
  - the direction d,
  - the bandelet coefficients $\{\langle f, b_\mu\rangle\}$.

Note that there might exist some dyadic squares in which we do not have a geometry because the square does not contains any geometric singularity. In those cases we simply keep the original wavelet coefficients.

Think in Terms of Number of Bits We denote by R the number of bits needed to both specify a bandelet basis $\mathscr{B} = \{b_\mu\}_\mu$ and code the coefficients of f in this basis. It can be decomposed into $R = \sum R_j = \sum (R_{jS} + R_{jG} + R_{jB})$, where, for each scale 2^j:

- R_{jS} is the number of bits needed to encode the dyadic segmentation. To code the quadtree we use 1 bit for each split.
- R_{jG} is the number of bits that code the optimal direction d in each square of width L using an arithmetic coder.
- R_{jB} is the number of bits needed to encode the quantized bandelet coefficients $Q_T(\langle f, b_\mu\rangle)$ using an arithmetic coder.

The function restored from its quantized bandelet coefficients is

$$f_R = \sum_\mu Q_T(\langle f, b_\mu\rangle)\, b_\mu \quad \text{with distortion} \quad \|f - f_R\|^2.$$

To find the best basis $\mathscr{B}$ for a given quantization step T, we minimize a Lagrangian $\mathscr{L}$ that can be shown to approximate the Lagrangian of the true distortion rate (see [Le Pennec and Mallat 2004]):

$$\mathscr{L}(f, R, \mathscr{B}) = \|f - f_R\|^2 + \lambda\, T^2 \sum_j (R_{Sj} + R_{Gj} + R_{Bj}).$$

The Quadtree Optimization Thanks to the additivity of the Lagrangian and the quadtree structure, the minimization of $\mathscr{L}$ can

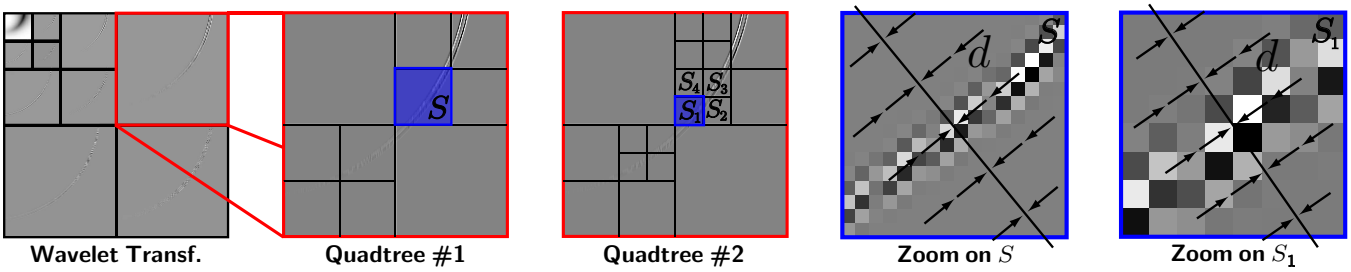


Figure 6: *Two examples of quadtree segmentations of the wavelet space. Each segmentation leads to a different bandelet bases.*

be performed using a fast bottom-up algorithm, this is the last step (10) of the bandelet transform algorithm.

Recall that in the previous steps (1)-(9), see figure 4, we have recorded, for each dyadic square S (of size less than $2^{-j/2}$) the value $\mathscr{L}(S) = \mathscr{L}(f,R,\mathscr{B})$ of the Lagrangian restricted to S, together with the best quantized direction d. Then, for each scale 2^j, we compute the quadtree structure:

- Initialize the quadtree: each smallest square S of width $L = 4$ pixels is a leaf, record the corresponding optimal geometry d, and initialize $\mathscr{L}_0$, the cumulative Lagrangian of the sub-tree, to $\mathscr{L}_0(S) = \mathscr{L}(S)$.
- Start with squares S of size $L = 8$ pixels.
- For each square S, we denote by (S_1,S_2,S_3,S_4) its 4 sub-squares,

  $$\text{and} \quad \mathscr{L}'(S) = \mathscr{L}_0(S_1) + \mathscr{L}_0(S_2) + \mathscr{L}_0(S_3) + \mathscr{L}_0(S_4) + \lambda T^2$$

  is the Lagrangian of the sub-tree (the additional λT^2 is due to the split cost $R_S = 1$ bit). The sub-squares should be merged if $\mathscr{L}(S) < \mathscr{L}'(S)$. If so, declare S as a leaf, record the optimal geometry d. Update $\mathscr{L}_0(S) = \min(\mathscr{L}(S), \mathscr{L}'(S))$.
- While $L < 2^{-j/2}$, do $L \leftarrow 2L$ and repeat the previous step.

This algorithm explore each dyadic square up to a size $2^{-j/2}$. A progressive refinement to search for the optimal d can avoid testing every possible direction, and leads to an overall complexity of $O(N^{5/4})$ for an image of N pixels.

Mathematical Result Replacing the 1D wavelet transform by a 1D Alpert multiwavelet transform [Alpert 1992] allows to prove the following result exposed in [Peyré and Mallat 2005b]. It is an extension of the results on the optimality of bandelet approximation [Le Pennec and Mallat 2005].

Given f a C^2-geometrically regular function, the transform coding f_R with R bits in the bandelet basis $\mathscr{B}$ minimizing $\mathscr{L}(f,R,\mathscr{B})$, with $R = R_S + R_G + R_B$, satisfies

$$\|f - f_R\|^2 \leqslant C R^{-2} \log^2(R),$$

with C a constant that depends only on the function f. We note the following important points:

- The bandelet approximation exponent -2 is optimal for C^2-geometrically regular functions.
- The reconstructed function is as regular as the original function.
- There is no blocking artifact due to the segmentation.

6 Application to Geometry Image and Normal Map Compression

Geometry Images Compression Geometry images [Gu et al. 2002] perform a completely regular remeshing of a 3D model so that it can be stored in an RGB image. The main issue with this approach is the large distortion induced by the planar mapping. However, multi-chart and spherical geometry images have been introduced to overcome this difficulty [Sander et al. 2003; Hoppe and Praun 2003]. We have chosen to use spherical geometry images, which limits our tests to genus-0 closed surfaces. The extension to multi-chart geometry images is possible but out of the scope of this paper.

The simplest way to compress a geometry image is to use an image transform (e.g. wavelets) with special boundary conditions (see [Hoppe and Praun 2003]). This is natural and follows the theoretical construction proposed in [Dahmen and Schneider 2000]. We measure the reconstruction error between a mesh $\mathscr{M}$ and its reconstruction with R bits $\mathscr{M}_R$ using

$$\text{PSNR}(\mathscr{M},\mathscr{M}_R) = 20\log_{10}(peak/d_H(\mathscr{M},\mathscr{M}_R)),$$

where *peak* is the bounding box diagonal and d_H is the RMS symmetric Hausdorff distance, computed using [Cignoni et al. 1998].

The main difference between the L^2 norm and the geometric distance d_H is that the latter is insensitive to warpings in parameter space, which do not change the shape of the surface. To remove this bias in coding, we use a simple fix by first performing a local change of coordinates (estimated using coarse scale approximation), and then allocating more bits (3x) for the details in the normal direction rather than in the tangential direction. This has proven useful in other compression schemes [Hoppe and Praun 2003; Guskov et al. 2000], but further theoretical studies remain to be done.

Figure 9 shows the Hausdorff distortion curves. Note that even for geometry images with blurred features (Gargoyle), there is still a PSNR improvement of over 1.4dB. On these geometry images, large distorsion is caused by the spherical mapping, which adds some artificial anisotropy. Figure 8 shows the Hausdorff distortion on small patches extracted from various surfaces. These geometry images are not corrupted with artificial geometry, but we still notice a PSNR improvement of over 1.5dB.

Normal Maps Compression A normal map is a color texture that encodes the normals of the surface on a regular grid. One usually renders a coarse mesh and adds the high-frequency geometric details of the surface using a normal map. In our tests we use normal maps that are either created by hand (see figure 1 and 9 for the generator models) or by using the methodology of spherical geometry images (see figure 9 for the other normal maps).

In our tests we encode the maps as RGB images, one channel per spacial axis, and we re-normalize the normal map after compression. The reconstruction error is measured using the traditional PSNR

$$\text{PSNR}(f, f_R) = 20\log_{10}(\|f\|_\infty / \|f - f_R\|_2).$$

Figure 9 shows the L^2 distortion curves, with a typical PSNR enhancement of $+2$dB for normal maps with strong geometrical features (Generator), and $+1.3$dB for normal maps with more smoothed features (Armadillo and Tira). These results clearly show the strength of our approach for normal map compression, which is more agressive than for geometry image compression. Some arguments can explain this fact:

- Wavelets perform reasonably well for geometry images, which only have discontinuities of tangents (decay exponent of $-3/2$).
- A normal map has strong fine scale geometric content where our bandelet transform performs well.

7 Conclusion

In this paper we have described a surface functional model that takes into account most geometric features present in CAD and scanned 3D surfaces. In this setting, the compression problem is well understood, and can be solved using harmonic analysis construction. We introduced second generation orthogonal bandelets that have several desirable properties for compression

- The construction is *orthogonal* which is important for compression.
- Basis functions are *regular* and hence introduce no blocking artifacts.
- It provides a *multiscale* representation of the geometry. This corresponds to the nature of most surfaces, and eases theoretical analysis.
- The transform coding rate is asymptotically optimal for the L^2 norm on C^2-geometrically regular functions.

Our numerical results show that bandelet bases provide a significant improvement over state of the art compression schemes for geometrically regular surfaces.

Acknowledgments The generator models are courtesy of CryTek. Spherical parameterizations and models are courtesy of H. Hoppe. We are greatly indebted to Rakan El-Khalil for his work in editing the paper.

References

AGARWAL, P. K., AND SURI, S. 1998. Surface approximation and geometric partitions. *SIAM Journal on Computing 27*, 4 (Aug.), 1016–1035.

AGARWAL, S., RAMAMOORTHI, R., BELONGIE, S., AND JENSEN, H. W. 2003. Structured importance sampling of environment maps. *ACM Trans. Graph. 22*, 3, 605–612.

ALLIEZ, P., AND GOTSMAN, C. 2005. *Recent Advances in Compression of 3D Meshes*. Springer-Verlag, 3–26.

ALLIEZ, P., COHEN-STEINER, D., DEVILLERS, O., LEVY, B., AND DESBRUN, M. 2003. Anisotropic polygonal remeshing. *ACM Transactions on Graphics. Special issue for SIGGRAPH conference*, 485–493.

ALPERT, B. 1992. *Wavelets and Other Bases for Fast Numerical Linear Algebra*. C. K. Chui, editor, Academic Press, New York.

BIERMANN, H., ZORIN, D., AND LEVIN, A. 2000. Piecewise smooth subdivision surfaces with normal control. In *Proc. of SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, 113–120.

CANDÈS, E., AND DONOHO, D. 1999. *Curvelets: A surprisingly effective nonadaptive representation of objects with edges*. Vanderbilt University Press.

CIGNONI, P., ROCCHINI, C., AND SCOPIGNO, R. 1998. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum 17*, 2 (June), 167–174.

COHEN-STEINER, D., ALLIEZ, P., AND DESBRUN, M. 2004. Variational shape approximation. *ACM Transactions on Graphics. Special issue for SIGGRAPH conference*, 905–914.

DAHMEN, W., AND SCHNEIDER, R. 2000. Wavelets on manifolds I: Construction and domain decomposition. *SIAM Journal on Mathematical Analysis 31*, 1 (Jan.), 184–230.

DANA, K. J., VAN GINNEKEN, B., NAYAR, N., AND KOENDERINK, J. J. 1999. Reflectance and texture of real-world surfaces. In *ACM Transactions on Graphics*, vol. 18, 1–34.

DAUBECHIES, I., RUNBORG, O., AND SWELDENS, W. 2004. Normal multiresolution approximation of curves. *Constructive Approximation 20*, 3, 399–463.

DEROSE, T., KASS, M., AND TRUONG, T. 1998. Subdivision surfaces in character animation. In *Proc. of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, 85–94.

DO, M. N., AND VETTERLI, M. 2005. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions Image on Processing, To appear*.

ECK, M., DEROSE, T., DUCHAMP, T., HOPPE, H., LOUNSBERY, M., AND STUETZLE, W. 1995. Multiresolution Analysis of Arbitrary Meshes. *Computer Graphics 29*, Annual Conference Series, 173–182.

FARIN, G. 1993. *Curves and Surfaces for Computer Aided Geometric Design*, 3. ed. Academic Press, Boston.

GARLAND, M., AND HECKBERT, P. 1997. Surface simplification using quadric error metrics. *Proc. of SIGGRAPH 1997*, 209–215.

GU, X., GORTLER, S., AND HOPPE, H. 2002. Geometry Images. *Proc. of SIGGRAPH 2002*, 355–361.

GUSKOV, I., VIDIMCE, K., SWELDENS, W., AND SCHRÖDER, P. 2000. Normal meshes. In *Proc. of SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, 95–102.

HERTZMANN, A., AND ZORIN, D. 2000. Illustrating smooth surfaces. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co., 517–526.

HOPPE, H., AND PRAUN, E. 2003. Shape compression using spherical geometry images. *Multiresolution in Geometric Modelling*.

HOPPE, H., DEROSE, T., DUCHAMP, T., HALSTEAD, M., JIN, H., MCDONALD, J., SCHWEITZER, J., AND STUETZLE, W. 1994. Piecewise smooth surface reconstruction. In *Proc. of SIGGRAPH 94*, Computer Graphics Proceedings, Annual Conference Series, 295–302.

HOPPE, H. 1996. Progressive meshes. *Proc. of SIGGRAPH 1996*, 99–108.

KHODAKOVSKY, A., AND GUSKOV, I. 2003. *Compression of Normal Meshes*. Springer-Verlag, In Geometric Modeling for Scientific Visualization.

LE PENNEC, E., AND MALLAT, S. 2004. Sparse Geometrical Image Approximation with Bandelets. *IEEE Transaction on Image Processing 14*, 4, 423–438.

LE PENNEC, E., AND MALLAT, S. 2005. Bandelet Image Approximation and Compression. *SIAM Multiscale Modeling and Simulation*, to appear.

LEVOY, M., PULLI, K., CURLESS, B., RUSINKIEWICZ, S., KOLLER, D., PEREIRA, L., GINZTON, M., ANDERSON, S., DAVIS, J., GINSBERG, J., SHADE, J., AND FULK, D. 2000. The digital michelangelo project: 3D scanning of large statues. In *Proc. of Siggraph 2000*, 131–144.

LINDSTROM, P., AND TURK, G. 1998. Fast and memory efficient polygonal simplification. *Proc. IEEE Visualization '98* (Oct.), 279–286.

MALLAT, S. 1998. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego.

MATEI, B., AND COHEN, A. 2002. *Nonlinear Subdivison Schemes : Applications to Image processing, in Tutorials on Multiresolution in Geometric Modelling*. Springer Verlag, 93–97.

OHTAKE, Y., BELYAEV, A., AND SEIDEL, S. 2004. Ridge-valley lines on meshes via implicit surface fitting. *ACM Transactions on Graphics 23*, 3 (Aug.), 609–612.

OWADA, S., NIELSEN, F., OKABE, M., AND IGARASHI, T. 2004. Volumetric illustration: Designing 3d models with internal textures. *Proceedings of SIGGRAPH 2004*, 322–328.

PEERCY, M., AIREY, J., AND CABRAL, B. 1997. Efficient bump mapping hardware. In *Proc. of SIGGRAPH 1997*, 303–306.

PEYRÉ, G., AND MALLAT, S., 2005. Bandelets toolbox, available on Matlab Central. `http://www.mathworks.com/matlabcentral/`.

PEYRÉ, G., AND MALLAT, S. 2005. Image approximation with geometric bandelets. In *Preprint CMAP*.

SANDER, P., WOOD, Z., GORTLER, S., SNYDER, J., AND HOPPE, H. 2003. Multi-chart Geometry Images. *Proc. Symposium on Geometry Processing 2003*, 146–155.

SCHRÖDER, P., AND SWELDENS, W. 1995. Spherical Wavelets: Efficiently Representing Functions on the Sphere. In *Proc. of SIGGRAPH 95*, 161–172.

SLABAUGH, G., CULBERTSON, B., MALZBENDER, T., AND SCHAFER, S. 2001. A survey of methods for volumetric scene reconstruction from photographs. In *Proc. of IEEE Eurographics Workshop*, Springer-Verlag, Wien, 81–100.

WAKIN, M., ROMBERG, J., CHOI, H., AND BARANIUK, R. 2005. Wavelet-domain Approximation and Compression of Piecewise Smooth Images. *IEEE Transactions on Image Processing, To appear*.

WANG, L., WANG, X., TONG, X., LIN, S., HU, S., GUO, B., AND SHUM, H.-Y. 2003. View-dependent displacement mapping. *ACM Trans. Graph. 22*, 3, 334–339.

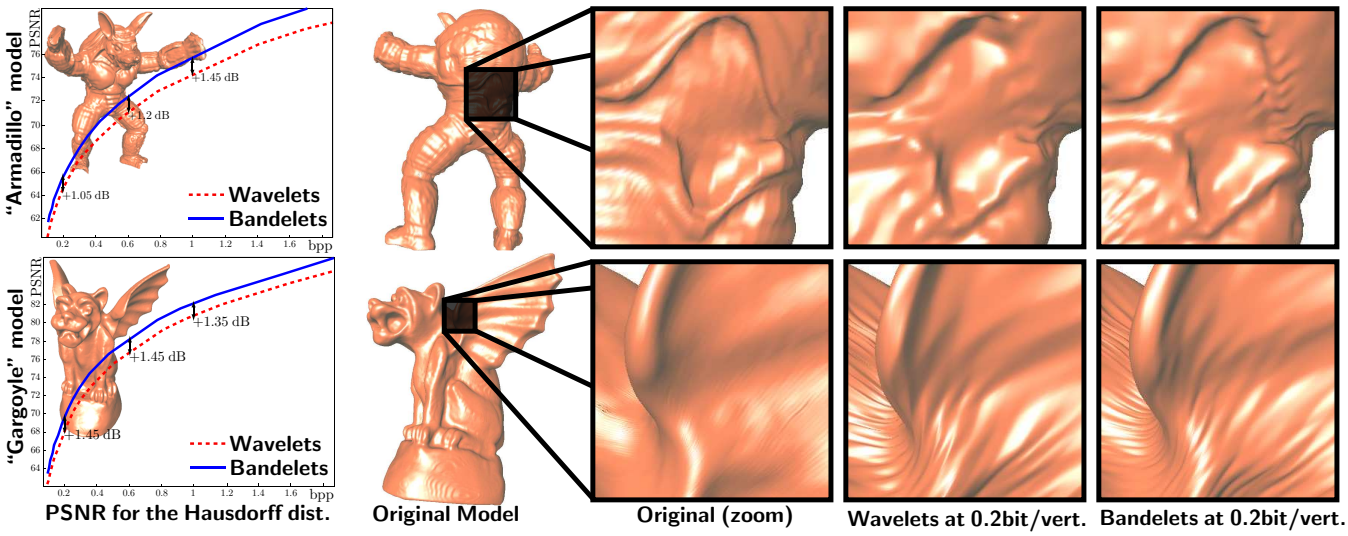


Figure 7: *Hausdorff distortion results for geometry images compression.*

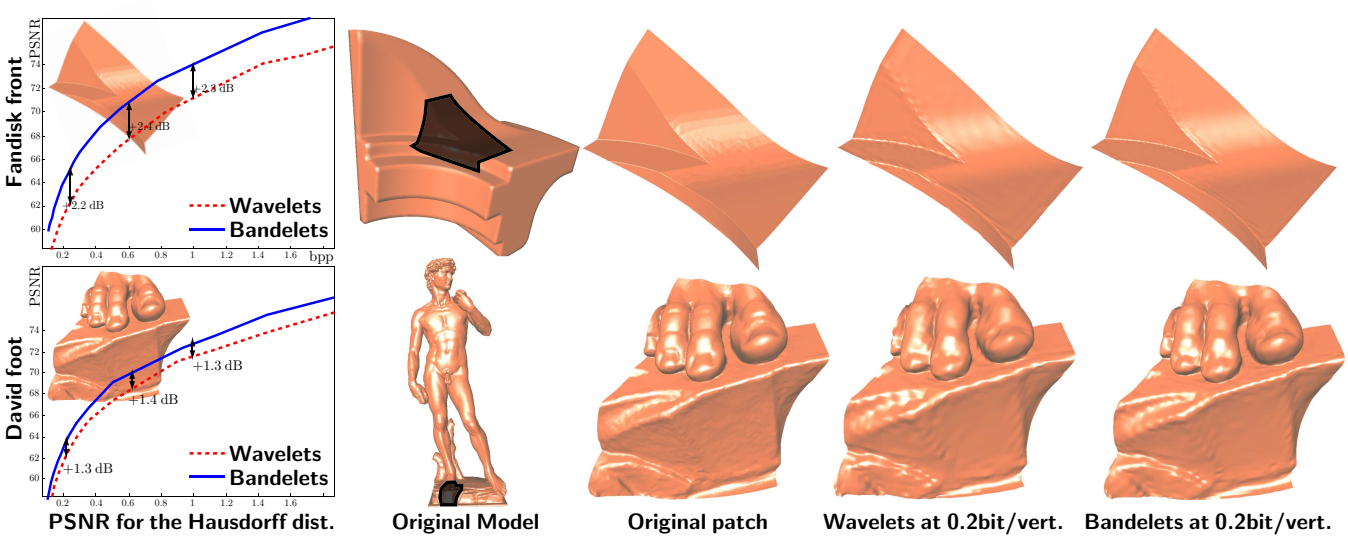


Figure 8: *Hausdorff distortion results for geometry images patches compression.*

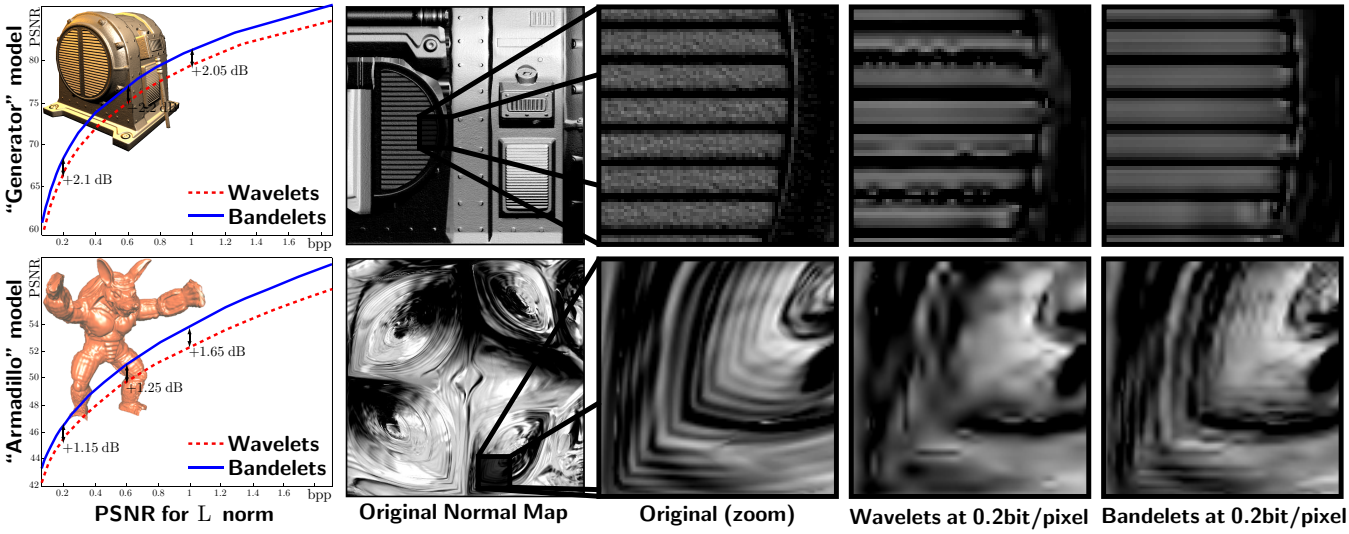


Figure 9: L^2 *distortion results for normal maps compression.*