Preprocessing for the decomposition of images with normal offsets

Ward Van Aerschot
Evelyne Vanraes
Adhemar Bultheel

Report TW 475, October 2006

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

Preprocessing for the decomposition of images with normal offsets

Ward Van Aerschot
Evelyne Vanraes
Adhemar Bultheel

Report TW475, October 2006

Department of Computer Science, K.U.Leuven

Abstract

The normal offset decomposition is a recent method to approximate images consisting of smoothly colored areas separated by smooth contours. In contrast to wavelet approximation methods, that perform suboptimally in this setting, this method is non-linear; the decomposition depends on the actual data. In every iteration new points are added by searching from the midpoint of the edges of the previous approximation along the normal direction until it pierces the surface that represents the image. The piercing points are attracted towards steep transitions and the edges that connect the new and old points line up against the contours in the image.
The normal offset algorithm starts from an initial triangulation of the rectangular domain of the image. The choice of these initial points and triangles determines the quality of the resulting approximation. The most straightforward choice for the initial triangulation is two triangles that share a diagonal. However, other choices that are adapted to the specific image are more efficient because piercing points will only be placed on, or in the neighborhood of a discontinuity if there is an edge crossing the contour on the previous level. In this paper we investigate a method to find an initial triangulation that also makes use of the piercing procedure inherent to the normal offset decomposition algorithm.

Keywords : normal meshes, multiresolution, image compression.
AMS(MOS) Classification : Primary : 94.A08, Secondary : 68.U10, 49.M27.

Preprocessing for the decomposition of images with normal offsets

Ward Van Aerschot, Evelyne Vanraes and Adhemar Bultheel

October 12, 2006

Abstract

The normal offset decomposition is a recent method to approximate images consisting of smoothly colored areas separated by smooth contours. In contrast to wavelet approximation methods, that perform suboptimally in this setting, this method is non-linear; the decomposition depends on the actual data. In every iteration new points are added by searching from the midpoint of the edges of the previous approximation along the normal direction until it pierces the surface that represents the image. The piercing points are attracted towards steep transitions and the edges that connect the new and old points line up against the contours in the image.
The normal offset algorithm starts from an initial triangulation of the rectangular domain of the image. The choice of these initial points and triangles determines the quality of the resulting approximation. The most straightforward choice for the initial triangulation is two triangles that share a diagonal. However, other choices that are adapted to the specific image are more efficient because piercing points will only be placed on, or in the neighborhood of a discontinuity if there is an edge crossing the contour on the previous level. In this paper we investigate a method to find an initial triangulation that also makes use of the piercing procedure inherent to the normal offset decomposition algorithm.

1 Introduction

Wavelets were originally developed to handle point singularities or discontinuities in one dimension. At each resolution level only a limited number of wavelets overlap with the discontinuity, hence the compact representation in terms of the wavelet basis functions. In two dimensions we have to deal with line singularities, for example contours in images. The number of wavelets needed to catch the steep transitions rises exponentially with the resolution level j as their support decays like 2^{-2j}. In the literature several methods have been proposed to overcome the weaknesses of wavelet methods and to catch the line singularities in a more efficient manner. There exist redundant linear decompositions using directional frames like ridgelets [7], curvelets [2] and contourlets [5]. Also highly nonlinear approximation schemes like wedgelets [6] and bandelets [10] have been explored.

In this paper we use a nonlinear approximation scheme referred to as normal approximation. This scheme has been investigated in the context of smooth curve approximation in the plane [3], and of smooth surface approximation in $\mathbb{R}^3$ [8]. It was first applied to piecewise smooth two dimensional functions with line discontinuities in [9], and applied to the specific setting of images in [11].

The normal offset decomposition method needs an initial triangulation of the rectangular domain of the image, and a corresponding base mesh, defined over that triangulation, that approximates the surface in three dimensional space that can be associated with the image. In each iteration we construct a finer mesh by adding vertices in between the vertices of the coarse mesh. We first predict the new vertices, for example linear prediction yields the midpoint between two old mesh points. Then we search in a direction normal to the coarse mesh starting from the prediction until we pierce the surface and choose the piercing point as the new vertex. Finally we store the difference between each prediction and the actual new vertex. This detail coefficient is called a normal offset.

In one dimension the new vertices are attracted towards point singularities, i.e. jumps in the function. The two dimensional method exploits this one dimensional property by constructing edges in such a way that they are attracted towards 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. Despite this interesting property the advantages of the normal offset decomposition algorithm become most visible once two or more vertices on a contour are found. Then the rate of approximation speeds up considerably.

In this paper we describe a preprocessing step that accelerates the initial process of finding vertices on the contours, such that less iterations are needed to approximate an image given a certain upperbound on the error. The paper is organized as follows. Section 2 and 3 review the concept of normal offsets and the extensions to digital images. Section 4 and 5 describe the proposed preprocessing step, and finally section 6 discusses the results.

2 Normal offsets

The concept of the normal offset method can best be illustrated in the one dimensional setting. Figure 1 shows an example with a discontinuity and two sample points $p_{j,k}$ and $p_{j,k+1}$ already found on the function, one on either side

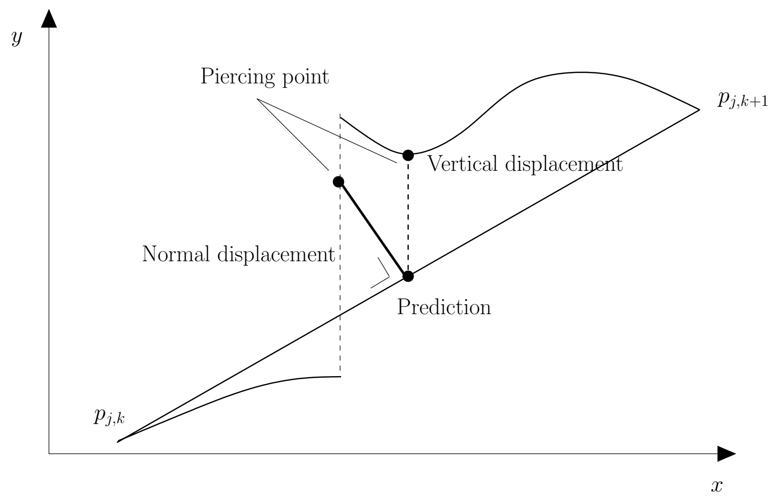

Figure 1: Conceptual comparison between vertical and normal offsets.

of the discontinuity. The polyline connecting the consecutive sample points is a coarse scale approximation of the function. The approximation is refined by adding new sample points in between the old points. To find a new sample location we first make a prediction, for example linear subdivision leads to the midpoint between two adjacent sample points. In the wavelet setting and other dyadic refinement strategies, the actual new point $p_{j+1,2k+1}$ is then the point on the function at the same location. The difference between the prediction and the actual point, that is used as wavelet coefficient, is a vertical offset with respect to the ordinate axis. In contrast, the normal offset method will choose the actual point on the curve by searching for a piercing point along a line normal to the coarse scale approximation. The detail coefficient is now a normal offset instead of a vertical offset, namely the signed distance between the prediction and the piercing point in the normal direction.

In the one dimensional case the piercing points are attracted towards point discontinuities, i.e. steep function transitions. In the two dimensional setting, we want to attract two dimensional structures, namely edges in the approximating mesh, towards line discontinuities. Moreover, we want the successive finer approximations to be hierarchical. 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 [4].

The above requirements are achieved as follows:

1. The prediction is the midpoint of the edge.
2. The normal ray is restricted to lie in a plane perpendicular to the domain.

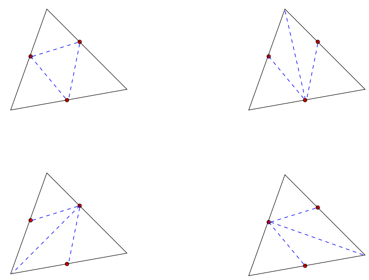

Figure 2: Different possible interconnections between new piercing points and old mesh points.

The restriction of the normal direction to lie in a vertical plane results in a edge refinement method. Newly found piercing points have their ordinates on the same edge the normal ray was shot from and the edges of each triangle are recursively subdivided. This ensures the hierarchical interconnection between successive approximations.

During the construction of a finer mesh, the new piercing points have to be connected with each other and the old mesh points to form new edges and triangles. Instead of fixing the connection rule in advance, we let it depend on the underlying data. The four possible interconnections between piercing points and old vertices are depicted in figure 2. This additional flexibility allows to discourage edges to cross a contour. The contour will be approximated by a polyline consisting of triangle edges.

3 Digital setting

Images can be seen as piecewise smooth two dimensional functions. Working with a finite set of elements in the domain (pixels) that are mapped onto a finite set of discrete values, introduces some problems. Therefore the algorithm needs to be adapted such that it also works for the digital setting. We give a summary of the changes here, and refer for more details and a formal description to [11].

1. The midpoint of an edge does not necessarily coincide with the center of a pixel. We rasterize the edges with the Bresenham algorithm [1]. This is a popular line rasterisation algorithm used in computer graphics. Every edge is now considered as a set of a neighboring pixel locations. We now choose the midpoint as the middle pixel between the outer edge pixels.

2. Rasterizing the subedges again with the Bresenham algorithm might result in a set of pixels for a subedge that is not a subset of the set of pixels of the parent edge. Therefore the pixels of a subedge are found by dividing the

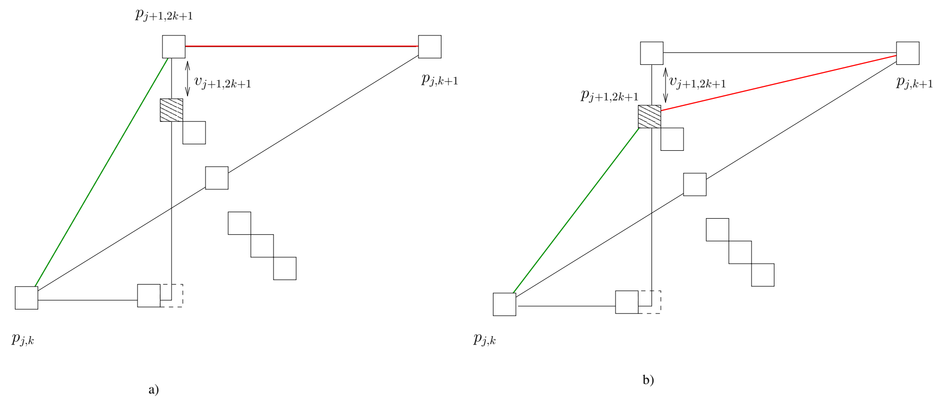

Figure 3: Also the normal ray is discretized. The vertical offset is the difference between the piercing point and the real function value at that location.

pixels of the parent edge between its two child edges and the Bresenham algorithm is only used for new edges that have no parent edge.

3. We also discretize the normal ray. It is possible that the piercing point does not coincide with a function value, but lies on the 'jump' between two function values. This is illustrated in figure 3. The difference between the piercing point $p_{j+1,2k+1}$ and the function value at the corresponding position is the vertical offset $v_{j+1,2k+1}$. The use of additional vertical offsets is needed to exactly reconstruct the image.

4 Motivation

Although practical examples learn that new piercing points are attracted towards discontinuities and that the normal ray will pierce the discontinuity at a certain level, the current research on the approximation rate concentrates on the case in the limit where piercing points on the singularity are already found. Moreover, the known convergence results only apply for the specific case of horizon images. These are gray value images consisting of several homogeneous areas separated by smooth contours [6].

In one dimension, after having located the singularity position, the algorithm breaks down into two independent approximations with a similar behavior left and right of the singularity. In two dimensions the singularity cannot be located exactly with a limited number of vertices. As the edges refine, we find successive piecewise linear approximations of the contours. The theoretical convergence rate is $\mathcal{O}(n)$ in the L_2 and L_1 norm with n the number of coefficients [9, 11].

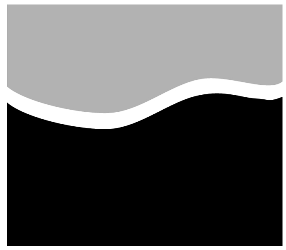

Figure 4: Original image for the examples.

In this paper we focus on the first stage of the algorithm, namely locating the singularities. Even with an arbitrary choice of initial vertices, the normal offset algorithm almost immediately starts placing new vertices close to the singularities because the normal direction tends to point to the singularities. If the two vertices of an edge lie on either side of a singularity, then the piercing point in the normal direction is always closer to the singularity than the piercing point in the vertical direction. Once the distance between two such vertices becomes smaller than the height of the singularity, the newly inserted point is always on the singularity itself. However, from experimental results we know that during this process the theoretical approximation rate is not achieved, and that it is actually the most difficult and slowest part of the algorithm [11].

To find a point on a contour, we need an edge in the coarse mesh that crosses it. If we start from an initial triangulation with few edges, only few contours can potentially be crossed and thus be found in that iteration. It takes several iterations to distribute new points and edges over the image such that all areas are covered.

Therefore we propose to start from a dense grid with more edges instead. When subsequently the normal offset decomposition algorithm is applied, many of the piercing points will coincide with a singularity. On the other hand, there will also be many piercing points that do not carry useful information, e.g. in flat regions. If we select only the first type, we can use them to construct a base mesh with many vertices placed on the main contours. In this way we still exploit the property that piercing points are attracted towards singularities, but avoid the need for many iterations to distribute the points over the image. In the next section we develop an algorithm based on this idea.

5 Base mesh generation

We illustrate the different steps of the algorithm with the picture shown in figure 4. It is a horizon image with two contours. The gray scale value, or height of the corresponding surface, is different on both sides of the contour.

For the initial dense triangulation, we choose a grid with K equidistant points (figure 8(1)). Practical examples show that starting from random initial points triangulated with e.g. the Delaunay algorithm gives equivalent results. The use of a triangulation algorithm however requires more computational effort. Hence, the choice for an equidistant grid.

After applying the normal offset algorithm once, we want to select only the piercing points on the discontinuities. We do not have a priori knowledge on the positions of the contours. It is therefore impossible to exactly pinpoint the piercing points that coincide with a singularity.

A piercing point has a normal offset and a vertical offset. The normal offset is the signed distance between the prediction and the piercing point, and the vertical offset is the signed distance between the piercing point and the real function value at that location. Based on the geometric information in the coarse mesh it is easy to compute the distance between the prediction and the real function value at the location of the piercing point. We call this the vector offset.

We want to use the information on the offsets to select relevant piercing points. The normal offset will be small when the prediction is already close to the singularity, and large otherwise. Consequently, there is no direct relation between the value of the normal offset and the close presence of a prominent contour. The vertical offset is needed to exactly reconstruct the image, and can only be non negligible when a discontinuity is pierced. When the normal rays pierce through smooth areas, the piercing points will be close to the function itself and the vertical offsets are small or zero. Unfortunately, even when the piercing point coincides with a discontinuity, the vertical offset can be small, depending on where exactly the discontinuity is pierced.

Figure 5 shows an example of a pierced discontinuity in two extreme situations. The dotted lines correspond to a coarse approximation and normal ray for which the the normal offset is maximal and the vertical offset is zero. The dashed lines on the other hand correspond to a zero normal offset and a maximal vertical offset. For situations in between these extremes the value of the normal offset varies between zero and r_2, the radius of the largest circle, and the value of the vertical offset varies between zero and r_1, the radius of the smallest circle. The value of the combined vector offset in contrast, can only take values between r_1 and r_2. Therefore we use the vector offset as criterium in the selection process.

Figure 6 shows plots of the piercing points with the largest normal, vertical and vector offsets respectively. The gray value of a point represents the value of the offset, with black indicating the largest offset at hand. On the plot of the normal offset we see points on the same contours with diverse values. The value depends on how far the contour lies from an edge in the initial dense

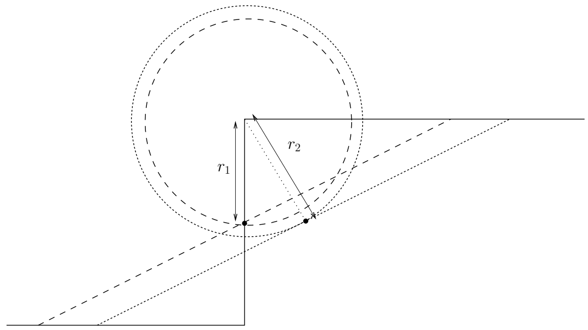

Figure 5: A piercing point on a singularity can have large and small normal and vertical offsets.

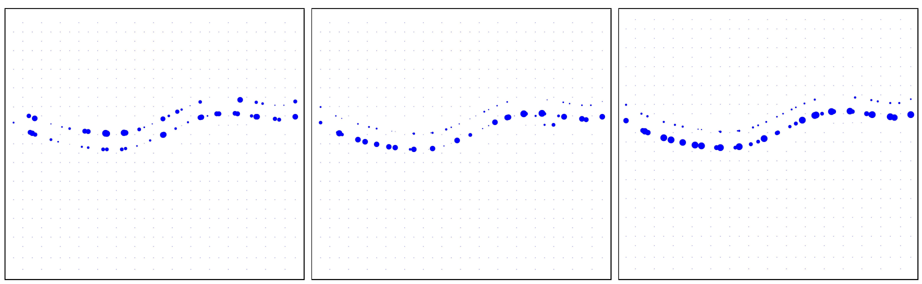

Figure 6: Piercing points with the largest normal, vertical and vector offsets respectively.

grid. When we use the vertical offsets, some points on the singularities are not detected because when the piercing point lies on the function itself, the vertical offset is zero. We selected the same number of points for the three plots, this leads to the selection of random points that also have vertical offset zero but that are not on the contours. The value of the vector offset, in contrast, varies only slightly and depends mainly on the height of the singularity.

We select the N piercing points with the largest vector offsets and also add the four corners of the image. Figure 7 shows a plot of the vector offsets ordered according to their magnitude. We see two abrupt changes that can be associated with the two contours of different height. The value of the normal offset turns out to be correlated with the height of the singularity at hand. This information can potentially be used to decide how many points should be selected. For now we assume that the user supplies the value for N.

In the next step we triangulate the selected points with the Delaunay algorithm. The resulting mesh (figure 8(2-5)) includes points on the most important contours of the image, but is too dense to yield a better approximation with

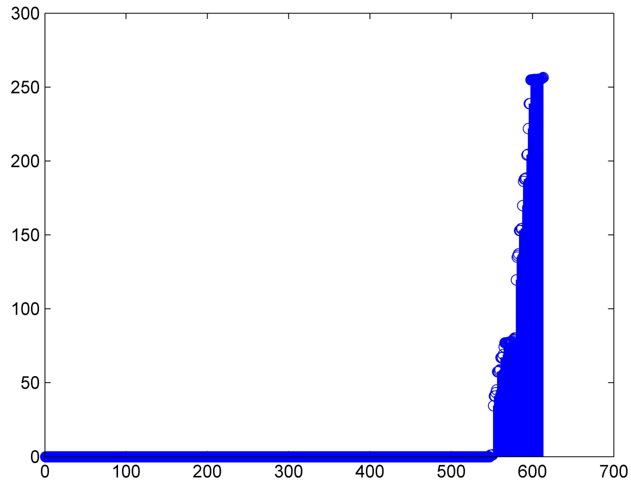

Figure 7: Magnitude of vector offsets.

the normal offset decomposition algorithm. It is possible to improve this base mesh with some straightforward additional steps. First we notice that the mesh contains many thin triangles with a common vertex on a singularity. We obtain a mesh with less unnecessary thin triangles by removing points adjacent to small angles until the minimal angle over all triangles is greater than a predefined minimal angle θ, but with the restriction that the total number of vertices cannot become smaller than a predefined number L. If more than L vertices are left we also remove those lying on a straight line by collapsing the edges.

From practical experiments we know that if two singularities are closer to each other than their heights, the approximation in the gap depends on the relation between the height h of, and the distance s between the singularities: it takes approximately h/s iterations to reach the bottom of the gap. To accelerate this process and skip these iterations we add an additional vertex between every two vertices of the base mesh so far. Finally we once more remove points until the smallest angle in the triangulation is larger than a certain angle α.

We now summarize the preprocessing algorithm that yields the base mesh. The numbers correspond to the illustrations in figure 8.

1. Initial configuration: dense equidistant mesh with K points
2. Do one step of the normal offset algorithm.
3. Select N piercing points with the largest vector offsets.
4. Add the four corners of the image.

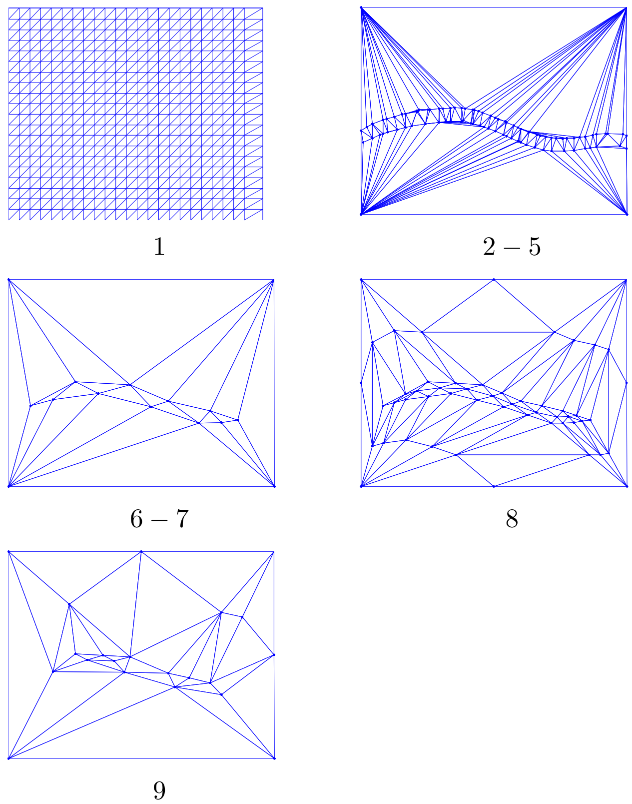

Figure 8: Illustration of the different stages in the preprocessing that leads to a base mesh that is suited as start for the normal offset decompostition algorithm.

5. Do a Delaunay triangulation.
6. Remove points until the minimal angle $\geq \theta$ or the number of vertices equals L.
7. If there are more than L points left: remove also those lying close to each other on a straight line.
8. Add the midpoints to all edges.
9. Remove points until the minimal angle $\geq \alpha$.

6 Results

We compare the results of the normal offset decomposition algorithm with and without the preprocessing step. In figure 9 the base mesh consists only of two triangles with a common edge and in figure 10 we used the base mesh found in the previous section for this example. Both approximations are compared in figure 11 that shows a plot of the Peak Signal to Noise Ratio with respect to the number of triangles in the approximation. The dashed line represents the approximation without base mesh, and the solid line represents the approximation starting from the base mesh generated with the proposed method.

Initially the decomposition without the base mesh yields a better approximation. The quality of the approximation with such a small number of triangles is however not sufficient for compression and other applications. At more accurate resolution levels the normal offset algorithm benefits from the placing of the initial triangles, the PSNR is higher. The naive decomposition starting from two triangles cannot achieve the same rate because the normal offset algorithm is a so-called "greedy" algorithm. It makes irrevocable decisions as it goes, once a point is inserted it cannot be moved of removed again. If certain details are not yet captured, the only solution is to add more points and triangles. When the initial base mesh is carefully chosen, this decreases the total number of triangles needed for a given quality.

7 Summary and further research

In this paper we proposed a preprocessing algorithm for the normal offset decomposition that yields a basemesh with vertices on the main contours. This enhances the quality of the resulting approximation. The algorithm uses several parameters, such as the number of vertices in the mesh at different stages and the minimal angle in the triangulation, that have to be supplied be the user. More research and practical experiments are needed to automatically detect the best value for this parameters for a specific image, so that the preprocessing can be incorporated in eg. a full functional compression module for images.

Acknowledgements

The work is partially 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. The scientific responsibility rests with the authors.

References

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

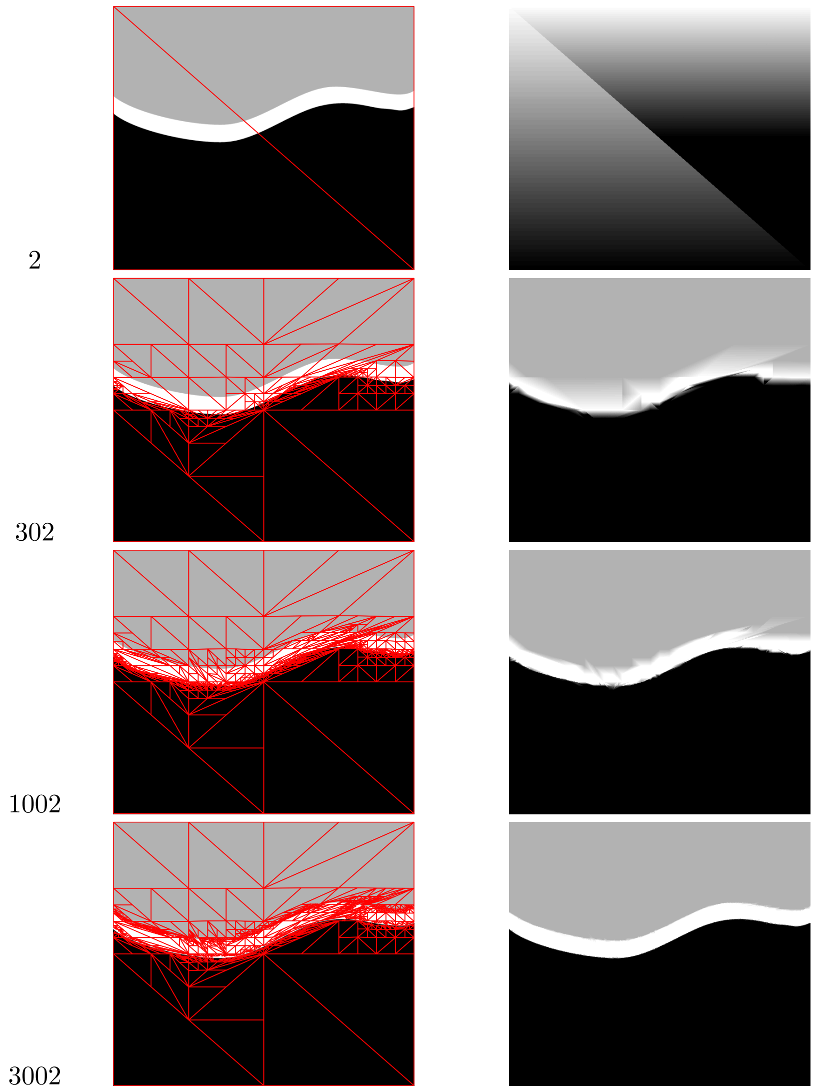

Figure 9: Different levels in the algorithm starting from a base mesh consisting of only two triangles.

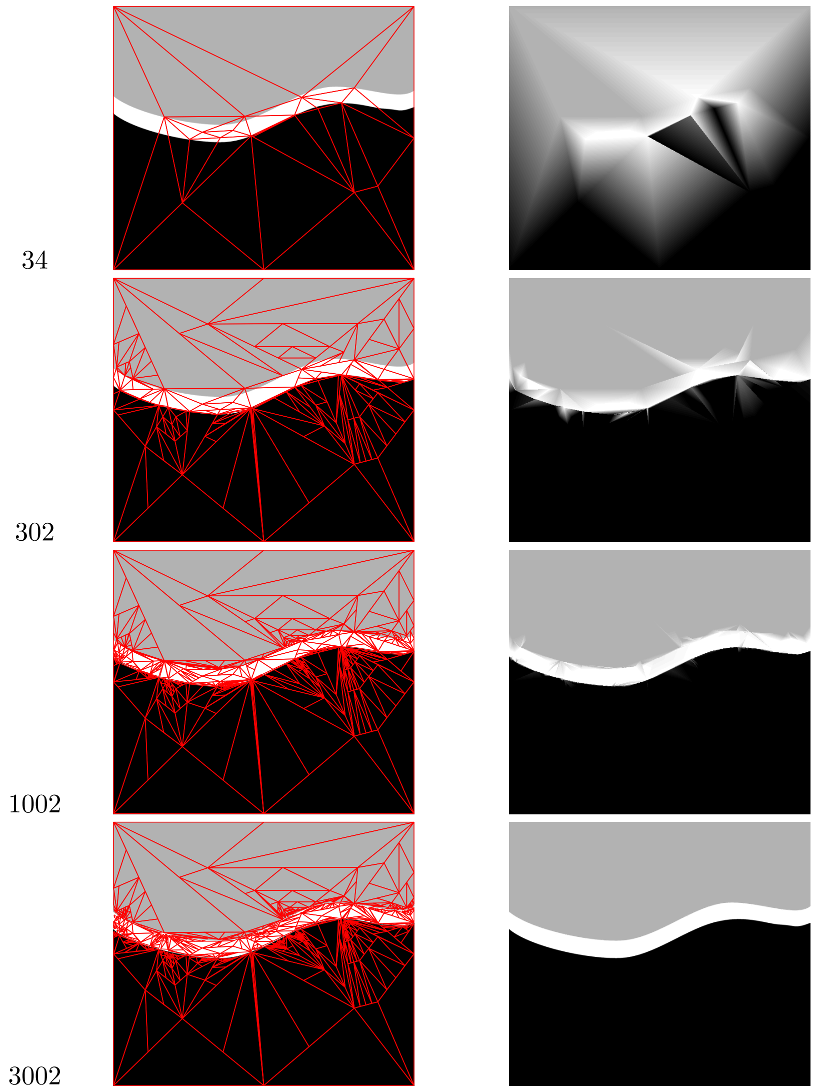

Figure 10: Different levels in the algorithm starting from the base mesh shown in figure 8.

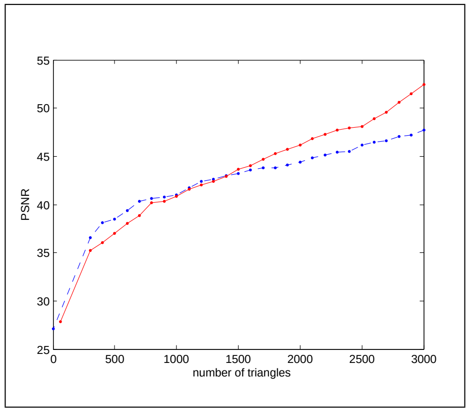

Figure 11: Peak Signal to Noise Ratio wrt the number of triangles in the approximation. Solid line: with base mesh. Dashed line: without base mesh.

[2] E. J. Candès and D. L. Donoho. Curvelets - a surprisingly effective non-adaptive representation for objects with edges. Technical report, Department of Statistics, Stanford University, 2000.

[3] I. Daubechies, O. Runborg, and W. Sweldens. Normal multiresolution approximation of curves. *Constructive Approximation*, 20(3):399–463, May 2004.

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

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

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

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

[8] I. Guskov, K. Vidimee, W. Sweldens, and P. Schröder. Normal meshes. In *SIGGRAPH 2000 Conference Proceedings*, pages 95–102, 2000.

[9] M. Jansen, R. Baraniuk, and S. Lavu. Multiscale approximation of piecewise smooth two-dimensional functions using normal triangulated meshes. *Appl. Comp. Harm. Anal.*, 19(1):92–130, 2005.

[10] E. Le Pennec and S. Mallat. Sparse geometric image representations with bandelets. *IEEE Trans. Image Proc.*, 14(4):423–438, 2005.

[11] W. Van Aerschot, M. Jansen, and A. Bultheel. Digital normal meshes in image compression. Technical Report TW 446, Department of Computer Science, K.U.Leuven, January 2006.