

Context-sensitive Filtering of Terrain Data based on Multi Scale Analysis

Steffen Goebbels and Regina Pohle-Fröhlich

Faculty of Electrical Engineering and Computer Science, Niederrhein University of Applied Sciences,
Reinarzstr. 49, 47805 Krefeld, Germany

Keywords: Surface Triangulation, Polygon Count Reduction, Wavelet Transform.

Abstract: This paper addresses triangle count reduction for regular terrain triangulations. Supported by additional information available from cadastre data, non-linear low pass filtering based on wavelet coefficients is applied. The aim is to generate a simplified view of a city model's terrain so that still landmarks can be easily recognized. Height data, that are not critical for visual impressions, are compressed with lower resolution than height data belonging to landmarks. Results are presented for the German city of Krefeld.

1 INTRODUCTION

Airborne laser scanning of terrain results in a huge amount of vertices that is not immediately suitable for drawing a surface triangulation. In this paper an approach is discussed where additional information from register data about ground usage is used for data reduction. From that information we derive precision that is necessary to draw certain areas. Goal is that the terrain still should be recognizable while the number of triangles is minimized.

Wavelet transforms are an often used method for point cloud compression, filtering and extraction of bare earth, cf. (Vu and Tokunaga, 2004; Wei and Bartels, 2006; Xu et al., 2007; Hu et al., 2014). Besides polygon merging, polygon elimination, vertex elimination, edge collapsing, and different retiling techniques, multi-resolution wavelet analysis is one of the main methods for mesh reduction, cf. (Kalbermatten, 2010, pp. 45–77), (Mocanu et al., 2011; Wünsche, 1998; Wiman and Yuchu, 2009). For example (Olanda et al., 2014; Bjørke and Nilsen, 2003; Bruun and Nilsen, 2003) give an overview of wavelet techniques for simplification of digital terrain models. These techniques do not use additional information besides height data. Studies were carried out with biorthogonal average interpolating wavelets. The property of conserving average height values at different scales yields good approximations of terrain.

As wavelet transform allows for local frequency filtering, it is predestined for local, context-sensitive data reduction. However, knowledge of local properties of ground is needed. For example, good visible sharp edges can be detected from height data using

curvature coefficients. Such edges should be maintained and not eliminated by filtering. In the example of a city model we do not have to detect such patterns in height data as we can use additional information from a geographic information system (GIS)¹. These cadastre data describe ground usage in terms of polygons that stand for streets, railway lines, bridges, etc. This approach to local filtering based on digital maps has been proposed before, for example in (Mahler, 2001, p. 58) in the context of coast lines. New is a set of heuristic rules to determine the local level of filtering using a simple filtering algorithm that differentiates between low frequencies describing terrain on a coarse scale and high frequencies needed for details.

Whereas for example in (Bjørke and Nilsen, 2003) one biorthogonal wavelet transform is used to filter the whole frequency spectrum, we apply two different transforms to minimize triangle count:

Local areas like embankments, shallow hills and heaps that afford a higher quality of rendering will be filtered using the simplest average interpolating wavelet that is the discontinuous Haar wavelet. We apply non-linear filtering using thresholds for wavelet coefficients. In our scenario the work with continuous orthogonal or biorthogonal wavelets is not necessary. In contrast, local support of the Haar wavelet helps to be very location specific.

Haar wavelet transform leads to artifacts like visible steps if only low frequencies remain after filtering. Therefore, areas which do not require detailed ren-

¹Case study is based on geographical data, which are protected by copyright: Geobasisdaten der Kommunen und des Landes NRW ©Geobasis NRW 2014.

dering will be low pass filtered via continuous linear spline approximation (that also can be interpreted as a biorthogonal wavelet transform based on B-splines). To be more specific, we do this spline based low pass filtering for the whole scenario and not only for areas with little detailed information. Reason is, that local areas with relevant details are not only detected heuristically on the basis of context information but also based on deviations between the surface of spline approximation and unfiltered heights. For these areas we triangulate with heights obtained from filtering with the Haar wavelet. Otherwise, heights for triangulation are taken from spline approximation. Spline approximation gives us a degree of freedom to further decrease triangle count while preserving a more detailed triangulation of areas that are displayed in higher quality. We modify the precision of spline interpolation to find a minimum number of triangles.

2 TRIANGULATION BASED ON FILTERED HEIGHTS

Our algorithm consists of two steps. In the first step, height data are filtered with spline approximation and wavelet transform as described in subsequent sections. In a second step, mesh simplification can be performed more efficient due to the adjusted heights. For the second step a variety of well established algorithms like edge collapsing can be used, cf. (Garland and Heckbert, 1997; Silva and Gomes, 2004). Also, the simplification can be made dependent on distance to the viewer by using geometry clipmaps as introduced in (Losasso and Hoppe, 2004). However, to test our approach we decided to use a very basic algorithm that is described in this section. The algorithm works with an equidistant mesh where 400 square kilometers are represented by $(5 \cdot 2^9 + 1)^2$ grid points. Triangulation is done for each square kilometer separately on the basis of squares that are split up into two triangles using a diagonal. We divide a square recursively into four smaller squares of equal size in case the surfaces of triangles deviate more than ϵ from previously filtered heights. A predefined accuracy level $\epsilon = 0.2$ m is a good compromise fitting with the accuracy of underlying laser scanning data, but we also show numbers for $\epsilon = 0.05$ and $\epsilon = 0.1$ m. Because mesh connectivity changes, larger ϵ might lead to visible gaps between surface triangles.

To avoid unsightly edges when displaying diagonal structures, we use following rule to choose the diagonal for splitting up a square into two triangles. We compute the four absolute differences between the arithmetic mean of all four heights of square vertices

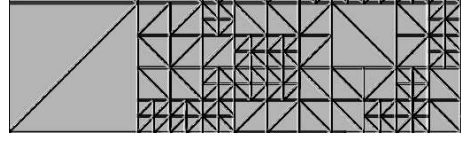


Figure 1: Triangulation based on an equidistant mesh.

and each of the four heights. If P is a vertex with a maximum difference, then we select the diagonal for which P is not end point. Further down, this rule is described with formula (1). Because no orientation of the diagonal is a-priori favored over the other, we do not get a type-1 triangulation for which this would be the case. A typical outcome is shown in Figure 1.

3 LINEAR SPLINE APPROXIMATION

Multi scale analysis on the basis of piecewise linear, continuous functions with local support seems to be adequate in connection with drawing triangulated surfaces. Therefore, in this section we reduce number of triangles with the means of a one-dimensional equidistant L^2 best linear continuous spline approximation on different scales. The result is a coarse model of terrain. Our approach to handle visible detailed structures is described in the next section. Pure edge collapsing methods like (Garland and Heckbert, 1997; Silva and Gomes, 2004) simplify the mesh in planar zones. Using spline approximation we significantly simplify in non planar zones as well. If such zones are not important for visual impression, we will not increase their level of detail later.

Using continuous, piecewise linear hat function

$$\Lambda(x) := \begin{cases} 1+x & : -1 \leq x < 0 \\ 1-x & : 0 \leq x \leq 1 \\ 0 & : x \notin [-1, 1] \end{cases}$$

we define scales $V_i, V_i \subset V_{i+1}$,

$$V_i := \left\{ \sum_{k=0}^{m \cdot 2^i} c_k \Lambda_{i,k}(x) : c_0, \dots, c_{m \cdot 2^i} \in \mathbb{R} \right\},$$

with $\Lambda_{i,k}(x) := m \cdot 2^i \Lambda(m \cdot 2^i \cdot x - k)|_{[0,1]}$ where “ $|_{[0,1]}$ ” denotes the restriction of a function to the real interval $[0, 1]$ and m is an odd natural number – in our case $m = 5$ because of $(5 \cdot 2^9 + 1)^2$ grid points.

Each scale has a B-spline basis of translated and scaled hat functions. They do not constitute an orthogonal set of functions but can be seen in context of biorthogonal wavelets. For the sake of simplicity, we directly compute best L^2 approximations and do not use terms of biorthogonal wavelets.

We simplify a function

$$f_{i+1} = \sum_{k=0}^{m \cdot 2^{i+1}} c_{i+1,k} \Lambda_{i+1,k}(x) \in V_{i+1}$$

to a coarser function $f_i = \sum_{k=0}^{m \cdot 2^i} c_{i,k} \Lambda_{i,k}(x) \in V_i$ by computing the best approximation with regard to L^2 inner product $\langle f, g \rangle := \int_0^1 f(x) \cdot g(x) dx$. It is well known, that coefficients $c_{i,k}$ form the unique solution of normal equations

$$\begin{bmatrix} \langle \Lambda_{i,0}, \Lambda_{i,0} \rangle & \dots & \langle \Lambda_{i,0}, \Lambda_{i,m \cdot 2^i} \rangle \\ \vdots & & \vdots \\ \langle \Lambda_{i,m \cdot 2^i}, \Lambda_{i,0} \rangle & \dots & \langle \Lambda_{i,m \cdot 2^i}, \Lambda_{i,m \cdot 2^i} \rangle \end{bmatrix} \cdot \begin{pmatrix} c_{i,0} \\ \vdots \\ c_{i,m \cdot 2^i} \end{pmatrix} = \begin{pmatrix} \langle f_{i+1}, \Lambda_{i,0} \rangle \\ \vdots \\ \langle f_{i+1}, \Lambda_{i,m \cdot 2^i} \rangle \end{pmatrix}.$$

By considering the representation of f_{i+1} as a linear combination of functions $\Lambda_{i+1,k}$, and by solving elementary integrals, we get

$$\frac{1}{6} \begin{bmatrix} 2 & 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & \dots & 0 & 0 & 0 \\ \vdots & & & & & & & \\ 0 & 0 & 0 & 0 & \dots & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 2 \end{bmatrix} \cdot \begin{pmatrix} c_{i,0} \\ \vdots \\ c_{i,m \cdot 2^i} \end{pmatrix} = \begin{bmatrix} \frac{5}{12} & \frac{1}{2} & \frac{1}{12} & 0 & 0 & 0 & 0 & \dots & 0 \\ \frac{1}{12} & \frac{1}{2} & \frac{5}{6} & \frac{1}{2} & \frac{1}{12} & 0 & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{12} & \frac{1}{2} & \frac{5}{6} & \frac{1}{2} & \frac{1}{12} & \dots & 0 \\ \vdots & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & \frac{1}{12} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & \frac{5}{12} \end{bmatrix} \cdot \begin{pmatrix} c_{i+1,0} \\ \vdots \\ c_{i+1,m \cdot 2^{i+1}} \end{pmatrix}$$

where the tridiagonal matrix on the left side belongs to $\mathbb{R}^{(m \cdot 2^{i+1}) \times (m \cdot 2^{i+1})}$, and the matrix on the right side is element of $\mathbb{R}^{(m \cdot 2^i) \times (m \cdot 2^{i+1})}$. Thus, we can start on scale n with a representation

$$f_n(x) = \sum_{k=0}^{m \cdot 2^n} c_{n,k} \Lambda_{n,k}(x)$$

where $c_{n,k} = a_k / (m \cdot 2^n)$ for $m \cdot 2^n + 1$ given height values $(a_0, a_1, \dots, a_{m \cdot 2^n})$. By solving normal equations we successively get coefficients $c_{i,k}$ that can be interpreted as heights $m \cdot 2^i c_{i,k}$ on the coarser grid of scale i , $0 \leq i < n$.

Tridiagonal systems of linear equations can be solved with Thomas-algorithm, cf. (Mooney and Swift, 1999, p. 235), so that we get a procedure APPROX($c_{i+1,0}, \dots, c_{i+1,m \cdot 2^{i+1}}$) that returns $(c_{i,0}, c_{i,1}, \dots, c_{i,m \cdot 2^i})$. Algorithm 1 shows how we apply this one-dimensional best approximation to all rows and then to columns of a matrix. This is similar to the simplified non-standard wavelet transform that will be discussed in Section 4. In order to ensure that

the result consists of heights and not of spline coefficients we have to normalize with factors $1/(m \cdot 2^{i+1})$ and $m \cdot 2^i$, respectively, for example

$$B[k, 0 : m \cdot 2^i] := m \cdot 2^i \cdot \text{APPROX}\left(\frac{1}{m \cdot 2^{i+1}} \cdot A_{i+1}[k, 0 : m \cdot 2^{i+1}]\right).$$

As best approximation is a linear calculation, factors can be shortened to $\frac{1}{2}$, cf. Algorithm 1.

Algorithm 1: Height data reduction. Height matrix $A_{i+1} \in \mathbb{R}^{(m \cdot 2^{i+1}) \times (m \cdot 2^{i+1})}$ is mapped to $A_i \in \mathbb{R}^{(m \cdot 2^i) \times (m \cdot 2^i)}$ using one-dimensional best approximations of rows and columns.

```

procedure HEIGHTDATAREDUCTION( $A_{i+1}$ )
  for  $k = 0 : m \cdot 2^{i+1}$  do
     $B[k, 0 : m \cdot 2^i] := \frac{1}{2} \cdot \text{APPROX}(A_{i+1}[k, 0 : m \cdot 2^{i+1}])$ 
  for  $k = 0 : m \cdot 2^i$  do
     $A_i[0 : m \cdot 2^i, k] := \frac{1}{2} \cdot \text{APPROX}(B[0 : m \cdot 2^{i+1}, k])$ 
  return  $A_i$ 
    
```

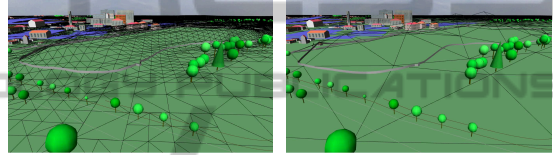


Figure 2: Reduction of triangles with three steps of linear spline approximation (triangles are transparent).

Computing of spline approximation consists of $n - i$ calls of Algorithm 1 so that we get a $(m \cdot 2^i) \times (m \cdot 2^i)$ -matrix A_i of filtered heights. Figure 2 shows the result of three calls in our example. The number $n - i$ of calls will be chosen with respect to the result of detail analysis such that the overall number of triangles becomes small. From A_i we derive approximate heights $h_i(j, k)$ for all $(m \cdot 2^n + 1) \times (m \cdot 2^n + 1)$ grid points (j, k) through interpolation. The given point (j, k) lies within the coarser net's square with vertices (x_0, y_0) , $(x_0 + 1, y_0)$, $(x_0, y_0 + 1)$, $(x_0 + 1, y_0 + 1)$ where x_0 and y_0 are down rounded integers defined by $x_0 := \lfloor j/2^{n-i} \rfloor$ if $j < m \cdot 2^n$ and $y_0 := \lfloor k/2^{n-i} \rfloor$ if $k < m \cdot 2^n$. For end points with $i = m \cdot 2^n$ set $x_0 := m \cdot 2^i - 1$, and for $k = m \cdot 2^n$ let $y_0 := m \cdot 2^i - 1$. In order to compute an approximate height with respect to scale i , we have to triangulate the square by choosing a diagonal as described in Section 2. Let

$$\bar{h} := (A_i[x_0, y_0] + A_i[x_0 + 1, y_0] + A_i[x_0, y_0 + 1] + A_i[x_0 + 1, y_0 + 1]) / 4$$

be the average of heights. If

$$\max\{|A_i[x_0, y_0] - \bar{h}|, |A_i[x_0 + 1, y_0 + 1] - \bar{h}|\} \leq \max\{|A_i[x_0 + 1, y_0] - \bar{h}|, |A_i[x_0, y_0 + 1] - \bar{h}|\} \quad (1)$$

then the diagonal connects (x_0, y_0) and $(x_0 + 1, y_0 + 1)$. Otherwise $(x_0, y_0 + 1)$ and $(x_0 + 1, y_0)$ are connected.

The surface of this triangulation at point (i, j) defines height $h_i(j, k)$. As a result, we get a coarse approximation of terrain that lacks detailed structures. Please also note that even in grid points the heights might be changed significantly by spline approximations (see figure 2). Compared to original data, these changes might lead to increased detail in some areas after the next processing steps (see lower left corner of pictures in Figure 3).

4 DISCRETE WAVELET TRANSFORM

We now deal with areas like heaps, bridge ramps and embankments that have to be shown in more detail. Despite higher accuracy is required, these areas also can be smoothed using a low pass filter based on discrete wavelet transform (DWT) for the Haar wavelet. In one dimension Algorithm 2 shows how to transform a vector $(a_0, \dots, a_{m \cdot 2^n - 1})$ (not dealing with last element $a_{m \cdot 2^n}$). In each step arithmetic means b_j of pairs of values a_{2j} and a_{2j+1} are computed. They represent data on a coarser scale. By adding wavelet coefficients $b_{j+N/2}$ to the mean value b_j , a_{2j} is reconstructed. Subtracting $b_{j+N/2}$ leads to a_{2j+1} . Wavelet coefficients can be interpreted as amplitudes of local frequencies. Haar scaling and wavelet functions form an orthonormal basis if correctly normed, cf. (Stollnitz et al., 1995). If wavelet coefficients are seen as scalar factors regarding this wavelet basis, then one has to divide by $\sqrt{2}$ instead of 2 in Algorithm 2 and gets coefficients $2^{-i/2}c_k$ as a result of Algorithm 2.

Algorithm 2: One-dimensional DWT for size $m \cdot 2^n$.

```

procedure STEP( $\vec{a}$ )
   $N := \text{SIZE}(\vec{a})$ 
  for  $j = 0 : N/2 - 1$  do
     $b_j := \frac{a_{2j} + a_{2j+1}}{2}, b_{j+N/2} := \frac{a_{2j} - a_{2j+1}}{2}$ 
  return  $\vec{b}$ 
procedure DWT( $\vec{a}, m, n$ )
  if  $n \leq 0$  then return  $\vec{a}$ 
   $\vec{b} := \text{STEP}(\vec{a})$ 
   $\vec{c}[0 : m \cdot 2^{n-1} - 1] := \text{DWT}(\vec{b}[0 : m \cdot 2^{n-1} - 1], m, n-1)$ 
   $\vec{c}[m \cdot 2^{n-1} : m \cdot 2^n - 1] := \vec{b}[m \cdot 2^{n-1} : m \cdot 2^n - 1]$ 
  return  $\vec{c}$ 

```

The one-dimensional transform often is extended to two dimensions by iteration. Another approach, that covers iteration as well but is not discussed here, is the use of multi dimensional wavelets. In the literature two iterative methods are prominent: standard and non-standard transform, cf. (Stollnitz et al., 1995). The standard transform is a tensor product

method that applies the one-dimensional transform to all rows of a matrix $A := [a_{j,k}]$ resulting in matrix B :

$$B = [\text{DWT}((a_{j,0}, a_{j,1}, \dots, a_{j,m \cdot 2^n - 1}))]_{j=0, \dots, m \cdot 2^n - 1}$$

In a second step each column of B gets transformed into a column $\text{DWT}((b_{0,k}, b_{1,k}, \dots, b_{m \cdot 2^n - 1, k}))$, $0 \leq k \leq m \cdot 2^n - 1$ of resulting matrix C . However, for our purpose of reducing height differences it appears to be a disadvantage of the standard transform that wavelet coefficients themselves get transformed during column transforms. For image manipulation instead of the standard wavelet transform often a non-standard variant is used that is slightly faster (but of same linear order). The non-standard transform iterates between row and column transformation steps and excludes previously computed wavelet coefficients in follow-on steps, see Algorithm 3.

Algorithm 3: Two-dimensional non-standard DWT.

```

procedure NST_DWT( $A$ )
   $m \cdot 2^n \times m \cdot 2^n := \text{SIZE}(A)$ 
  if  $n \leq 0$  then return  $A$ 
  for  $j = 0 : m \cdot 2^n - 1$  do
     $B[j, 0 : m \cdot 2^n - 1] := \text{STEP}(A[j, 0 : m \cdot 2^n - 1])$ 
  for  $j = 0 : m \cdot 2^n - 1$  do
     $\triangleright$  Use  $0 : m \cdot 2^{n-1} - 1$  for simplified transform.
     $C[0 : m \cdot 2^n - 1, j] := \text{STEP}(B[0 : m \cdot 2^n - 1, j])$ 
   $D[0 : m \cdot 2^{n-1} - 1, 0 : m \cdot 2^{n-1} - 1] :=$ 
     $\text{NST\_DWT}(C[0 : m \cdot 2^{n-1} - 1, 0 : m \cdot 2^{n-1} - 1])$ 
   $D[m \cdot 2^{n-1} : m \cdot 2^n - 1, 0 : m \cdot 2^n - 1] :=$ 
     $C[m \cdot 2^{n-1} : m \cdot 2^n - 1, 0 : m \cdot 2^n - 1]$ 
   $D[0 : m \cdot 2^{n-1} - 1, m \cdot 2^{n-1} : m \cdot 2^n - 1] :=$ 
     $C[0 : m \cdot 2^{n-1} - 1, m \cdot 2^{n-1} : m \cdot 2^n - 1]$ 
  return  $D$ 

```

Even with the non-standard transform, there is one column transformation step applied to previously computed coefficients of rows. Therefore, we choose to use a simplified version of non-standard transform that has been described in (Kopp and Purgathofer, 1998). The only change to Algorithm 3 is that the second loop stops at $m \cdot 2^{n-1} - 1$ instead of $m \cdot 2^n - 1$. The number of column transforms then only is half the number of row transforms. The advantage is that height differences immediately become apparent in terms of coefficients. The associated inverse computation for a single component $a_{j,k}$ reads:

$$\begin{aligned}
 a_{j,k} = & c_{\lfloor j2^{-n} \rfloor, \lfloor k2^{-n} \rfloor} \\
 & + \sum_{i=0}^{n-1} [c_{m \cdot 2^i + \lfloor j2^{i-n} \rfloor, \lfloor k2^{i-n} \rfloor} \Psi(j2^{i-n} - \lfloor j2^{i-n} \rfloor) \\
 & + c_{\lfloor j2^{i+1-n} \rfloor, m \cdot 2^i + \lfloor k2^{i-n} \rfloor} \Psi(k2^{i-n} - \lfloor k2^{i-n} \rfloor)]
 \end{aligned} \tag{2}$$

where ψ is the Haar wavelet function

$$\Psi(x) := \begin{cases} 1, & 0 \leq x < \frac{1}{2}, \\ -1, & \frac{1}{2} \leq x < 1, \\ 0, & \text{else} \end{cases}$$

that determines whether wavelet coefficients have to be added or subtracted.

Non-linear low pass filtering consists of cutting off higher summands depending on coefficients $c_{m \cdot 2^i + \lfloor j2^{i-n} \rfloor, \lfloor k2^{i-n} \rfloor}$ and $c_{\lfloor j2^{i+1-n} \rfloor, m \cdot 2^i + \lfloor k2^{i-n} \rfloor}$ during reconstruction of height data with equation (2). To this end we need context-sensitive sequences of thresholds. For a point (j, k) let $(t_0, t_1, \dots, t_{n-1})$ be such a sequence of non-negative numbers. We investigate two variants of non-linear filtering:

Variant 1: Let

$$\begin{aligned} \tilde{a}_{j,k} &:= c_{\lfloor j2^{-n} \rfloor, \lfloor k2^{-n} \rfloor} & (3) \\ &+ \sum_{i=0}^{n-1} f_{j,k} \left(c_{m \cdot 2^i + \lfloor j2^{i-n} \rfloor, \lfloor k2^{i-n} \rfloor}, i \right) \cdot \psi(j2^{i-n} - \lfloor j2^{i-n} \rfloor) \\ &+ \sum_{i=0}^{n-1} f_{j,k} \left(c_{\lfloor j2^{i+1-n} \rfloor, m \cdot 2^i + \lfloor k2^{i-n} \rfloor}, i \right) \cdot \psi(k2^{i-n} - \lfloor k2^{i-n} \rfloor) \end{aligned}$$

be a component of the filtered matrix where depending on point (j, k) and frequency i

$$f_{j,k}(x, i) := \begin{cases} x & : |x| \geq t_i \\ 0 & : |x| < t_i. \end{cases}$$

This variant allows to eliminate intermediate frequencies while higher frequencies might remain. Because high frequencies have a higher impact on number of triangles, we compare this approach with

Variant 2: Tails of reconstruction sum in equation (2) (i. e. high frequencies) get cut off. Let M_1 be the set of indices $i \in \{0, \dots, n-1\}$ with

$$\left| c_{\lfloor j2^{i+1-n} \rfloor, m \cdot 2^i + \lfloor k2^{i-n} \rfloor} \right| \geq t_i$$

and M_2 be the subset of $\{0, \dots, n-1\}$ where

$$\left| c_{m \cdot 2^i + \lfloor j2^{i-n} \rfloor, \lfloor k2^{i-n} \rfloor} \right| \geq t_i.$$

Then we cut off by using maximum values $m_1 := \max(M_1 \cup \{-1\})$ and $m_2 := \max(M_2 \cup \{-1\})$:

$$\begin{aligned} \tilde{a}_{j,k} &:= c_{\lfloor j2^{-n} \rfloor, \lfloor k2^{-n} \rfloor} & (4) \\ &+ \sum_{i=0}^{\max\{m_1-1, m_2\}} c_{m \cdot 2^i + \lfloor j2^{i-n} \rfloor, \lfloor k2^{i-n} \rfloor} \cdot \psi(j2^{i-n} - \lfloor j2^{i-n} \rfloor) \\ &+ \sum_{i=0}^{\max\{m_1, m_2\}} c_{\lfloor j2^{i+1-n} \rfloor, m \cdot 2^i + \lfloor k2^{i-n} \rfloor} \cdot \psi(k2^{i-n} - \lfloor k2^{i-n} \rfloor). \end{aligned}$$

If we alternate between reconstruction steps for columns and rows beginning with highest frequency

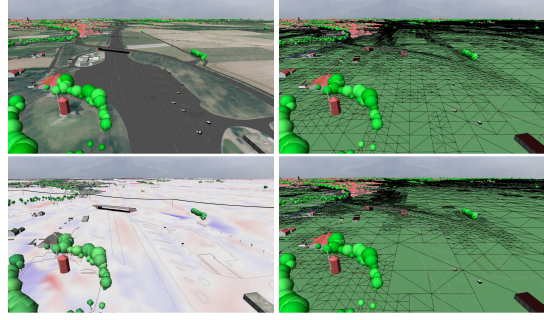


Figure 3: First row: unfiltered height data of a motorway, last row: filter (4) applied with $\epsilon = 0.2$ m and $\delta = 4.0$ m where height deviations are marked with colors in left picture: The bigger the difference, the darker the color, red marks points where unfiltered heights are larger than filtered heights, blue indicates the opposite.

$n-1$ going down to 0 then the formula starts adding summands when encountering a first coefficient that exceeds its threshold.

In contrast to JPEG 2000 we do not quantize wavelet coefficients. For triangle count reduction quantization does not appear to work as well as omission of coefficients. All non-zero coefficients of Haar wavelets define height alterations that might lead to additional triangles. But quantization does reduce storage space and transmission bandwidth, see (Olanda et al., 2014).

5 HEURISTICS BASED LOW PASS FILTERING

Our aim is to preserve salient structures like ramps, surroundings of bridges, hills, and shore lines. Polygons covering areas of these structures are available from GIS data. Whereas fields can be drawn with reduced precision, a more detailed drawing is needed for streets and railway lines. This section deals with heuristics to implement context-sensitive filtering that takes care of such differences.

If a difference of unfiltered height and the result of spline approximation at a point (j, k) is less than 0.3 m, then this difference is not disturbing. The threshold of 0.3 m is a parameter different than the accuracy ϵ that is used for mesh simplification on the basis of previously filtered heights. If the difference exceeds 0.3 m, in certain circumstances we compute a more specific height using low pass wavelet transform. In this case, depending on terrain usage and height of (j, k) , we divide coordinates (j, k) into three main classes:

1. Class 1: A point (j, k) is near to a point P , iff P is located within a square covering 9^2 grid points

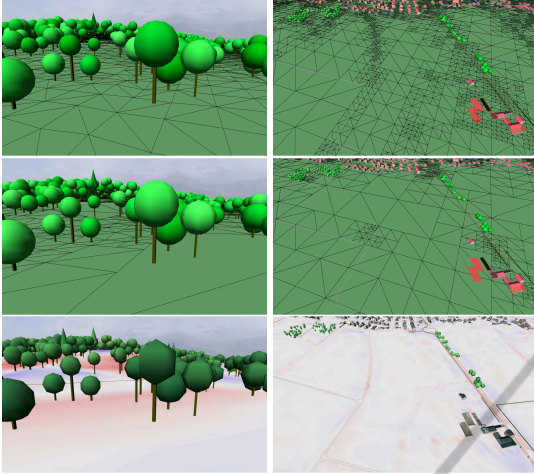


Figure 4: Triangulation based on unfiltered data (first row) and based on non-linear filtered data (second row) with filter (4) and $\epsilon = 0.2$ m, $\delta = 4.0$ m. Last row shows differences in the same manner as in Figure 3.

with (j, k) at its center. The size of this neighborhood was determined experimentally. Smaller squares lead to poorer quality whereas larger neighborhoods increase number of triangles unnecessarily. In our context, high precision is required for those points (j, k) that are near to a point P which fulfills one of following conditions.

- P is next to a shore line of a river or lake.
 - P is part of a railway embankment. If (j, k) is an interior point of the embankment then even extra precision is needed to avoid non smooth railway tracks.
 - P lies within the border polygon of a bridge. Surroundings of bridges need to be accurate so that bridges fit into terrain. In our example, even the finest triangle size is too coarse to avoid gaps between bridges and their ramps. This problem is fixed by extending bridges.
 - P lies within the polygon of a street that intersects at least one polygon of a bridge. Such streets can be part of ramps.
2. Class 2: Terrain of our city model is flat apart from some heaps and local hills. Therefore, zones outside class 1 with unfiltered heights that differ more than a parameter δ from the heights of spline approximation should be drawn with intermediate precision. Hence heap contours remain clearly visible. If a forest grows on a hill then even less precision might be needed as trees hide the ground. In our example, hilly terrain covered by trees is limited, so that special treatment of forests does not improve triangle count significantly. It is not performed.

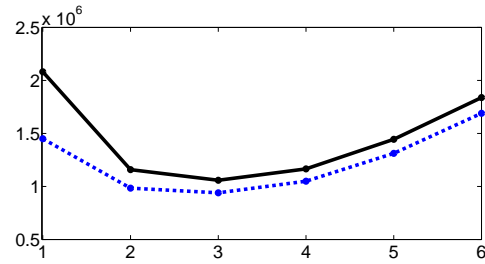


Figure 5: Triangle count for different numbers of spline iterations and accuracy levels ϵ , see Table 3. Broken line: $\epsilon = 0.2$ m, solid line: $\epsilon = 0.05$ m ($\delta = 4.0$ m, filter (4)).

3. Class 3: All remaining terrain can be drawn with poor precision, i.e., heights are taken directly from spline approximation.

We apply the simplified non-standard wavelet transform for the grid with $m \cdot 2^n = 5 \cdot 2^9$ points. Low pass filtering for points of classes 1 and 2 is done using specific sets of threshold values $(t_0, t_1, \dots, t_{n-1})$. Height values resulting from low pass filtering replace heights computed by spline approximation only if their distance to spline heights exceeds 0.3 m (less can't be seen).

For $\delta = 4$ m good looking results were obtained with threshold values from Table 1. We determined the values experimentally. Instead, one could define a distance measure between exact and filtered heights (for example an L^2 distance) together with separate local error bounds for areas of class 1 and class 2 as constraints. Then separately for both classes, variation of thresholds can yield a set of values for which the number of triangles becomes minimal.

Our data set contains an error where a hole appears in the middle of the harbor - probably an artifact from a ship. Generally, heights of grid points within polygons defining areas of water need to be constant, because water surfaces should be flat. Linear spline approximation and wavelet filtering do not guarantee this outcome. This is achieved by assigning an average value to all grid points of water polygons in a post-processing step.

6 RESULTS

We count triangles but we do not present a global measure for quality of filtered terrain. Because we allow to over-simplify areas that are uncritical for visual impressions, measures like scene-wide L^2 metrics that involve only height differences between filtered and unfiltered heights are not sufficient. A comparison of contour lines as done in (Bjørke and Nilsen, 2003)

Table 1: Thresholds.

coefficient	0	1	2	3	4	5	6	7	8
class 1	0	0.1	0.1	0.2	0.2	0.2	0.3	0.3	0.4
class 1, inner points of railway embankments	0	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
class 2	0	0.1	0.2	0.2	0.3	0.4	0.5	0.5	0.5
class 2, forest	0	0.1	0.2	0.2	0.3	20	20	20	20

Table 2: Triangle count for $\delta = 4$ m and three iterations of spline approximation.

triangulation accuracy level ϵ	triangles test data (no filtering)	triangles spline approx. with unfiltered detail	triangles, detail filtered with (3)	triangles, detail filtered with (4)
0.2 m	2901422	1221938	962012	940214
0.1 m	4564400	1405856	1046510	1035626
0.05 m	6144422	1434602	1067888	1058636

Table 3: Influence of spline approximation on triangle count for $\delta = 4$ m and wavelet filter (4), see Figure 5.

spline iterations	accuracy level $\epsilon = 0.05$ m		accuracy level $\epsilon = 0.2$ m	
	6144422 triangles without compression	triangles spline approximation	2901422 triangles without compression	triangles spline approximation
		triangles filtered detail		triangles filtered detail
1		2081120	1201376	1450412
2	544388	1158722	420140	984206
3	170210	1058636	135488	940214
4	48218	1165586	44372	1049498
5	12422	1444874	12308	1312340
6	3164	1838468	3140	1690142

might be a better tool but it lacks the ability to distinguish between “important” and “unimportant”.

Table 2 shows numbers of triangles in our case study. Last two columns correspond to results of filtering as described. Compression rates up to a factor of 5.8 can be reached. We have added the middle column in order to demonstrate positive effects of low pass Haar wavelet filtering of detailed areas. It shows triangle count when using exact heights instead of heights resulting from wavelet filtering of points belonging to class 1 and 2. Heights of all other points still are taken from spline approximation. The comparison of the two filter variants (3) and (4) shows that (4) produces slightly better results. That is no surprise as highest non-zero coefficients determine size of triangles. Non-zero coefficients of frequencies in the middle do not have this effect.

Finally, we have to determine a good value for the number of iterations of spline approximation. This number in turn depends on the rendering of detailed areas. Too many iterations lead to larger areas where height deviations exceed δ . The number of triangles saved through spline approximation then is outnumbered by the amount of additional triangles needed to show details. On the other hand, too few iterations generate a too detailed spline approximation and too

many triangles. Analysis of detailed areas then becomes obsolete. In our setting, for $\delta = 4$ m the lowest triangle count is obtained with three iterations, see Table 3 and Figure 5. The smaller δ the larger are areas of class 2 and the less iterations of spline approximation can be used for simplification. How δ influences the number of spline approximations and triangle count is shown in Table 4 and Figure 6.

With respect to different landscapes, the outcome of our approach to filtering is shown in Figures 3 and 4.

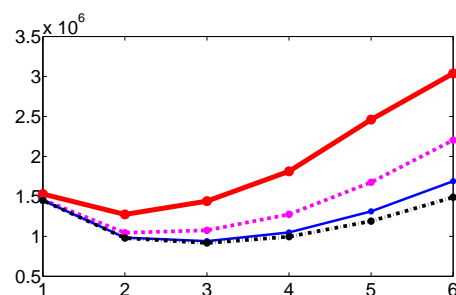


Figure 6: Triangle count depending on iterations of spline approximation for different height parameters δ (filter (4), $\epsilon = 0.2$ m), see Table 4. Thick solid line: $\delta = 1$ m, dashed line: $\delta = 2$ m, thin solid line: $\delta = 4$ m, dash-dot line: $\delta = 6$ m.

Table 4: Triangle count depending on iterations of spline approximation for different height parameters δ (filter (4), $\epsilon = 0.2$ m), see Figure 6.

iterations \rightarrow	1	2	3	4	5	6
$\delta = 1.0$	1529450	1274276	1440734	1813544	2460890	3038612
$\delta = 2.0$	1458482	1044968	1075772	1275776	1676342	2203256
$\delta = 4.0$	1450412	984206	940214	1049498	1312340	1690142
$\delta = 6.0$	1450058	978740	917552	995828	1190306	1489028

7 CONCLUSIONS AND FUTURE WORK

Whereas the MP3 algorithm compresses music data by utilizing a psychoacoustic model that defines accuracy of coefficients of a discrete cosine transform, we follow a similar but more elementary approach to reduce triangle count: Visually less important areas are lossy compressed by applying a continuous best spline approximation. More important areas are low pass filtered with a wavelet transform. Thresholds for wavelet coefficients depend on rules that take ground usage (cadastre data) and height differences into account. This is our “psycho visual” model. Results are promising. However, it should be investigated if an even better quality can be obtained by additional smoothing of certain surfaces like streets. Future work is required to explore the method in connection with other sets of terrain data in order to see whether experimentally determined thresholds still work well. The mesh simplification algorithm of Section 2 has been chosen for simplicity. Further work is needed to evaluate the filtering technique as a pre-processing step that generates input for established simplification methods.

REFERENCES

- Bjørke, J. and Nilsen, S. (2003). Wavelets applied to simplification of digital terrain models. *International Journal of Geographical Information Science*, 17:601-621.
- Bruun, B. and Nilsen, S. (2003). Wavelet representation of large digital terrain models. *Computers & Geosciences*, 29:695-703.
- Garland, M. and Heckbert, P. (1997). Surface simplification using quadric error metrics. In *Proc. 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 209-216.
- Hu, H., Ding, Y., Zhu, Q., Wu, B., Lin, H., Du, Z., Zhang, Y., and Zhang, Y. (2014). An adaptive surface filter for airborne laser scanning point clouds by means of regularization and bending energy. *ISPRS Journal of Photogrammetry and Remote Sensing*, 92:98-111.
- Kalbermatten, M. (2010). *Multiscale analysis of high resolution digital elevation models using the wavelet transform*. EPFL Lausanne, Lausanne.
- Kopp, M. and Purgathofer, W. (1998). Interleaved dimension decomposition: A new decomposition method for wavelets and its application to computer graphics. In *2nd International Conference in Central Europe on Computer Graphics and Visualisation*, pages 192-199.
- Losasso, F. and Hoppe, H. (2004). Geometry clipmaps: Terrain rendering using nested regular grids. *ACM Trans. Graph.*, 23(3):769-776.
- Mahler, E. (2001). *Scale-Dependent Filtering of High Resolution Digital Terrain Models in the Wavelet Domain*. University of Zurich, Zurich.
- Mocanu, B., Tapu, R., Petrescu, T., and Tapu, E. (2011). An experimental evaluation of 3d mesh decimation techniques. In *International Symposium on Signals, Circuits and Systems*, pages 1-4.
- Mooney, D. and Swift, R. (1999). *A Course in Mathematical Modeling*. The Mathematical Association of America, Washington D.C.
- Olanda, R., Pérez, M., Orduña, J., and Rueda, S. (2014). Terrain data compression using wavelet-tiled pyramids for online 3d terrain visualization. *International Journal of Geographical Information Science*, 28:407-425.
- Silva, F. and Gomes, A. (2004). Normal-based simplification algorithm for meshes. In *Proc. Theory and Practice of Computer Graphics (TPCG 04)*, pages 211-218.
- Strollnitz, E., DeRose, T., and Salesin, D. (1995). Wavelets for computer graphics: a primer, part 1. *IEEE Computer Graphics and Applications*, 15:77-84.
- Vu, T. and Tokunaga, M. (2004). Filtering airborne laser scanner data: A wavelet-based clustering method. *Photogrammetric Engineering & Remote Sensing*, 70:1267-1274.
- Wei, H. and Bartels, M. (2006). Unsupervised segmentation using gabor wavelets and statistical features in lidar data analysis. In *18th International Conference on Pattern Recognition*, pages 667-670.
- Wiman, H. and Yuchu, Q. (2009). Fast compression and access of lidar point clouds using wavelets. In *Urban Remote Sensing Event*, pages 1-6.
- Wünsche, B. (1998). *A Survey and Evaluation of Mesh Reduction Techniques*. University of Auckland, Auckland.
- Xu, L., Yang, Y., Jiang, B., and Li, J. (2007). Ground extraction from airborne laser data based on wavelet analysis. In *Proc. MIPPR/SPIE*.