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 Yu chu, 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.

1Case study is based on geographical data, which are protected by copyright: Geobasisdaten der Kommunen und des Landes NRW ©Geobasis NRW 2014.
Therefore, areas which do not require detailed rendering 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 \( \varepsilon \) from previously filtered heights. A predefined accuracy level \( \varepsilon = 0.2 \) m is a good compromise fitting with the accuracy of underlying laser scanning data, but we also show numbers for \( \varepsilon = 0.05 \) and \( \varepsilon = 0.1 \) m. Because mesh connectivity changes, larger \( \varepsilon \) 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 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

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

we define scales \( V_i \), \( V_j \subset V_{j+1} \),

\[
V_i := \left\{ \sum_{k=0}^{m-2} c_k \Lambda_{i,k}(x) : c_0, \ldots, c_{m-2} \in \mathbb{R} \right\},
\]

with \( \Lambda_{i,k}(x) := m \cdot 2^i \Lambda(m \cdot 2^i \cdot x - k) \mid_{[0,1]} \) where “\( \mid_{[0,1]} \)” 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

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

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

$$\begin{bmatrix}
<\Lambda_{0,0} \Lambda_{0,1} \cdots \Lambda_{0,2^i-1}> \\
<\Lambda_{1,0} \Lambda_{1,1} \cdots \Lambda_{1,2^i-1}> \\
\vdots \\
<\Lambda_{m-2^i,0} \Lambda_{m-2^i,1} \cdots \Lambda_{m-2^i,2^i-1}>
\end{bmatrix}
\begin{bmatrix}
\cdots \\
\cdots \\
\cdots
\end{bmatrix}
= \begin{bmatrix}
<\Lambda_{0,0} > \\
<\Lambda_{1,0} > \\
\vdots \\
<\Lambda_{m-2^i,0} >
\end{bmatrix}$$

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

$$\begin{bmatrix}
2 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\
1 & 4 & 1 & 0 & \cdots & 0 & 0 & 0 \\
0 & 1 & 4 & 1 & \cdots & 0 & 0 & 0 \\
\vdots \\
0 & 0 & 0 & 0 & \cdots & 1 & 4 & 1 \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
c_{i,0} \\
c_{i,1} \\
c_{i,2^i} \\
\vdots \\
c_{i,2i-1} \\
c_{i,2^i+2i-1} \\
\end{bmatrix}
= \begin{bmatrix}
\cdots \\
\cdots \\
\cdots
\end{bmatrix}$$

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

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

where $c_{n,k} = a_k/(m \cdot 2^n)$ for $m \cdot 2^n + 1$ given height values $(a_0, a_1, \ldots, a_{m \cdot 2^n})$. 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($c_{i+1,0}, \ldots, c_{i+1,m-2^i+1}$) that returns ($c_{i,0}, c_{i,1}, \ldots, c_{i,m-2^i+1}$). 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

$$\begin{align*}
B[k,0:m-2^i] := m \cdot 2^i \cdot \text{APPROX}(\frac{1}{m \cdot 2^{i+1}} A_{i+1}[k,0:m-2^{i+1}])
\end{align*}$$

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

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

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 $(m \cdot 2^i + 1) \times (m \cdot 2^i + 1)$ 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 $i = m \cdot 2^i$ set $x_0 := m \cdot 2^i - 1$, and for $k = m \cdot 2^i$ 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

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

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.
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 \( \{a_0, \ldots, a_{2^n-1}\} \) (not dealing with last element \( a_{2^n} \)). 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 \( b_{j+1/2} \) to the mean value \( b_j \), a wavelet is reconstructed. Subtracting \( b_{j+1/2} \) 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. (Strohleitner 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 \( 2^{-1/2}c_k \) as a result of Algorithm 2.

**Algorithm 2** One-dimensional DWT for size \( m \cdot 2^n \).

```plaintext
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}$)$

$c[0: m \cdot 2^n-1] := \text{DWT}($\vec{b}$[0: m \cdot 2^n-1], m, n-1)$

$c[m \cdot 2^n-1 : m \cdot 2^n-1] := \vec{b}[m \cdot 2^n-1 : m \cdot 2^n-1]$ 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. (Strohleitner 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 \( B \):

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

In a second step each column of \( B \) gets transformed into a column \( \text{DWT}((b_{0,k}, b_{1,k}, \ldots, b_{m \cdot 2^n-1,k})) \), \( 0 \leq k \leq m \cdot 2^n-1 \) 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.

```plaintext
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
        $B[j,0:m \cdot 2^n-1] := \text{STEP}($\vec{a}$[j,0:m \cdot 2^n-1])$
    for $j = 0 : m \cdot 2^n-1$ do
        \( \text{Use: } 0 : m \cdot 2^n-1 \) for simplifed transform.
        $C[0:m \cdot 2^n-1,j] := \text{STEP}($\vec{a}$[0:m \cdot 2^n-1,j])$
        $D[0:m \cdot 2^n-1,0:m \cdot 2^n-1] :=$
        $\text{NST_DWT}($\vec{a}$[0:m \cdot 2^n-1,0:m \cdot 2^n-1])$
        $D[m \cdot 2^n-1 : m \cdot 2^n-1,0:m \cdot 2^n-1] :=$
        $C[m \cdot 2^n-1 : m \cdot 2^n-1,0:m \cdot 2^n-1]$
        $D[0:m \cdot 2^n-1,1:m \cdot 2^n-1] :=$
        $C[0:m \cdot 2^n-1,1:m \cdot 2^n-1,0:m \cdot 2^n-1]$
    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 \( m \cdot 2^n-1 \) instead of \( m \cdot 2^n-1 \). The number of columns 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:

\[
a_{j,k} = c_{\left\lceil j/2^n\right\rceil,|k|2^{-n}} + \sum_{l=0}^{n-1} c_{\left\lceil m \cdot 2^{l+1} - \left\lfloor \left\lceil j/2^n \right\rceil \cdot 2^{l+n}\right\rfloor\right\rceil,|k|2^{-n}} \Psi(j/2^{l+n} - \left\lfloor j/2^{l+n} \right\rfloor)
\]

where \( \Psi \) 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 \( c_{m 2^i + |j2^{i-n}|, |k2^{i-n}|} \) and \( c_{j2^{i-1-n}, m 2^i + |k2^{i-n}|} \) during re-
construction of height data with equation (2). To this end we need context-sensitive sequences of th-
resholds. For a point \((j, k)\) let \((t_0, t_1, \ldots, t_{n-1})\) be such a sequence of non-negative numbers. We investigate two variants of non-linear filtering:

**Variant 1:** Let
\[
\tilde{a}_{j,k} := c_{|j2^{i-n}|, |k2^{i-n}|}
\]
\[
+ \sum_{i=0}^{n-1} f_{j,k} \left( c_{m 2^i + |j2^{i-n}|, |k2^{i-n}|}, i \right) \cdot \psi(j2^{i-n} - |j2^{i-n}|)
\]
\[
+ \sum_{i=0}^{n-1} f_{j,k} \left( c_{|j2^{i-1-n}|, m 2^i + |k2^{i-n}|}, i \right) \cdot \psi(k2^{i-n} - |k2^{i-n}|)
\]
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 \(i \in \{0, \ldots, n-1\}\) with
\[
\left| c_{|j2^{i-1-n}|, m 2^i + |k2^{i-n}|} \right| \geq t_i
\]
and \(M_2\) be the subset of \(\{0, \ldots, n-1\}\) where
\[
\left| c_{m 2^i + |j2^{i-n}|, |k2^{i-n}|} \right| \geq t_i
\]
Then we cut off by using maximum values \(m_1 := \max(M_1)\) and \(m_2 := \max(M_2)\):
\[
\tilde{a}_{j,k} := c_{|j2^{i-n}|, |k2^{i-n}|}
\]
\[
\max\{m_1, m_2\}
\]
\[
+ \sum_{i=0}^{n-1} c_{m 2^i + |j2^{i-n}|, |k2^{i-n}|} \cdot \psi(j2^{i-n} - |j2^{i-n}|)
\]
\[
\max\{m_1, m_2\}
\]
\[
+ \sum_{i=0}^{n-1} c_{|j2^{i-1-n}|, m 2^i + |k2^{i-n}|} \cdot \psi(k2^{i-n} - |k2^{i-n}|)
\]
If we alternate between reconstruction steps for columns and rows beginning with highest frequency \(n-1\) 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).

**Figure 3:** First row: unfiltered height data of a motorway, last row: filter (4) applied with \(\varepsilon = 0.2\) 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.

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

## 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 \(\varepsilon\) that is used for mesh simplification on the basis of previously filtered heights. If the difference...
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 (Björke and Nilsen, 2003) 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

Figure 5: Triangle count for different numbers of spline iterations and accuracy levels $\epsilon$, see Table 3. Broken line: $\epsilon = 0.2$ m, solid line: $\epsilon = 0.05$ m ($\delta = 4.0$ m, filter (4)).
### Table 1: Thresholds.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 1, inner points of railway embankments</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class 2</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Class 2, forest</td>
<td>0.0</td>
<td>0.1</td>
<td>0.2</td>
<td>0.3</td>
<td>0.4</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Triangulation accuracy level $\varepsilon$</th>
<th>Triangles test data (no filtering)</th>
<th>Triangles spline approx. with unfiltered detail</th>
<th>Triangles, detail filtered with (3)</th>
<th>Triangles, detail filtered with (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2 m</td>
<td>2901422</td>
<td>1221938</td>
<td>962012</td>
<td>940214</td>
</tr>
<tr>
<td>0.1 m</td>
<td>4564400</td>
<td>1405856</td>
<td>1046510</td>
<td>1035626</td>
</tr>
<tr>
<td>0.05 m</td>
<td>6144422</td>
<td>1434602</td>
<td>1067888</td>
<td>1058636</td>
</tr>
</tbody>
</table>

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 $\delta$. 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 $\delta$ the larger are areas of class 2 and the less iterations of spline approximation can be used for simplification. How $\delta$ 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.

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

<table>
<thead>
<tr>
<th>spline approximations</th>
<th>accuracy level $\varepsilon = 0.05$ m</th>
<th>accuracy level $\varepsilon = 0.2$ m</th>
</tr>
</thead>
<tbody>
<tr>
<td>triangles</td>
<td>6144422 triangles without compression</td>
<td>2901422 triangles without compression</td>
</tr>
<tr>
<td>iterations</td>
<td>triangles spline approximation</td>
<td>triangles filtered detail</td>
</tr>
<tr>
<td></td>
<td>triangles</td>
<td>triangles spline approximation</td>
</tr>
<tr>
<td></td>
<td>triangles</td>
<td>triangles filtered detail</td>
</tr>
<tr>
<td>1</td>
<td>1820300</td>
<td>1201376</td>
</tr>
<tr>
<td>2</td>
<td>544388</td>
<td>420140</td>
</tr>
<tr>
<td>3</td>
<td>170210</td>
<td>135488</td>
</tr>
<tr>
<td>4</td>
<td>48218</td>
<td>44372</td>
</tr>
<tr>
<td>5</td>
<td>12422</td>
<td>12308</td>
</tr>
<tr>
<td>6</td>
<td>3164</td>
<td>3140</td>
</tr>
</tbody>
</table>

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

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

REFERENCES


8