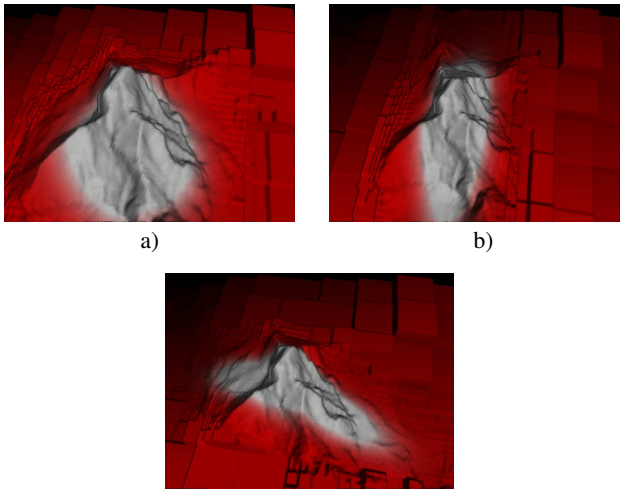
$$\boldsymbol{R}^m = \begin{pmatrix} \frac{\cos\Theta}{\sigma_x^m} & \frac{-\sin\Theta}{\sigma_x^m} & \frac{-x_0^m}{\sigma_x^m} \\ \frac{\sin\Theta}{\sigma_y^m} & \frac{\cos\Theta}{\sigma_y^m} & \frac{-y_0^m}{\sigma_y^m} \\ 0 & 0 & 1 \end{pmatrix} \tag{13}$$

and $\boldsymbol{p}^m = (x^m, y^m, 1)^T$ denotes a position in homogeneous coordinates.

To summarize this section: the control of the local level–of–detail of the wavelet approximation can be accomplished by using one single Gaussian weighting function which surrounds the region of interest in the initial data set. This Gaussian can be interpreted as a filter which is transformed into multiple Gaussians, one in each frequency channel in wavelet space. Premultiplying the coefficients with these Gaussian maps forces any subsequent thresholding to pass only coefficients located within the selected ROI. All others will be removed and hence the reconstructed signal will be most accurate within the ROI along with a Gaussian smoothing of the boundaries. Figure 5 stresses the effect of level–of–detail filtering. The digital terrain model is decomposed with Haar wavelets and filtered with Gaussians of different locations and parameters. The model is perfectly reconstructed within the focus of the Gaussian, whereas only the scaling functions represent the data outside. In the boundary region, less and less high frequency information is provided and the data becomes more and more "boxlike". Obviously, the proposed wavelet space filter acts as a *"magnifying glass"* onto the data.

We recommend applying the Gaussian filter and the thresholds only to the wavelets and keeping all coefficients of the scaling function, because they carry the DC fraction of the signal.



a) b) c)

Fig. 5: Example for filtering in wavelet space: Decomposition of Mount Matterhorn with Haar wavelets and filtering with different Gaussian space–frequency filters.
a) σ_x=25, σ_y=25, Θ=0. b) σ_x=45.5, σ_y=14.75, Θ=0.
c) σ_x=12.5, σ_y=50, Θ=0.217.

3. Quadtree Meshing

3.1. Point Removal in Regular Triangle Meshes

So far, we elaborated some mathematical criteria for approximating a surface data set, sampled on a regular grid, using a multiresolution hierarchy. In order to build an adaptive surface triangulation, however, it is necessary to remove unimportant mesh vertices and to find a triangulation of the remaining ones. The basic criterion, by which a mesh vertex is labeled as unimportant is given by the mathematical framework of the wavelet transform. Keeping in mind that any triangulation of the surface provides a planar approximation, we only have to bound the error between the original surface function $f(x,y)$ and the bilinear interpolant provided by a triangle. Supposing furthermore that the initial data is expanded by wavelet bases, the detail signal in iteration m helps us to decide whether or not each 2^m+1th mesh vertex is necessary for the triangle approximation. First, we visit each second vertex and analyze the value of the detail signal of iteration m=1. If, let's say, the detail signal $\Delta^1 f$ in some neighborhood of vertex n is sufficiently low, then the vertex is not important and the approximation can be accomplished by a linear interpolation between vertex n–1 and n+1. This scheme can now be applied recursively by subsequent computation of the detail signals $\Delta^m f, m=1,..,M$ and by visiting all dyadic vertices at positions $n = 2^m k+1$. Once the detail signal is sufficiently small and the adjacent vertices in step m–1 are already removed, we are allowed to label the current vertex as well.

As a consequence, our procedure results in recursively building a quadtree representation of the initial mesh by removing dyadic vertices. Fig. 6 again illustrates the thinning method which finally figures out the symbolic quadtree representation of the mesh vertices depicted as an example in Fig. 7. The nodes of the quadtree contain either pointers to some child–nodes, or in case of leaves, point to the entries of a vertex list.

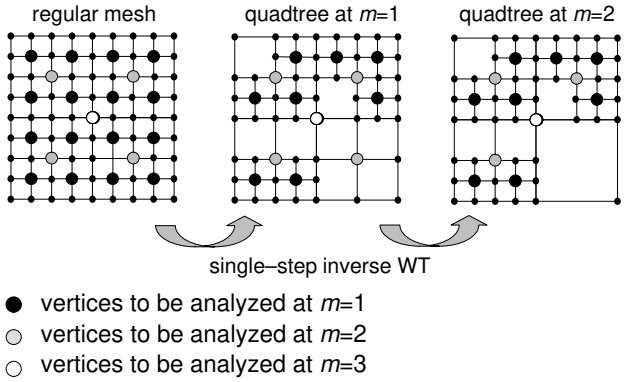


Fig. 6: Recursive growth of a quadtree from the regular mesh by analyzing the detail signals of the WT at each dyadic vertex.

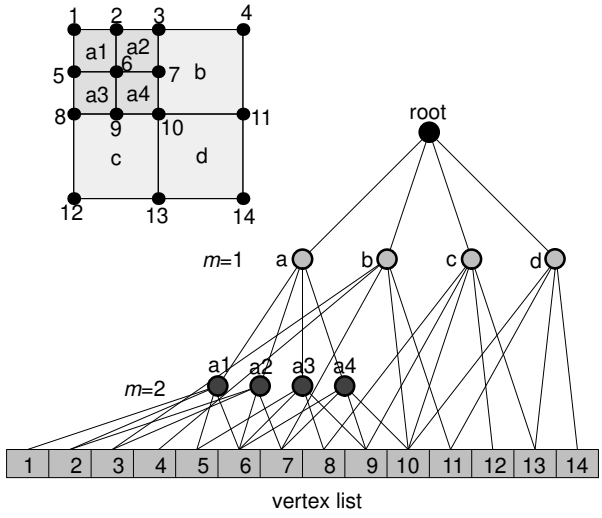


Fig. 7: Symbolic representation of the mesh using a quadtree data structure.

Note again here that the growth of our quadtree is entirely controlled by a single energy threshold τ embedded in the function