

# QEM-Filtering: A New Technique For Feature-Sensitive Terrain Mesh Simplification

F. Löffler and H. Schumann

University of Rostock / VC<sup>2</sup>G, Germany

---

## Abstract

Traditionally, terrain simplification is time consuming and generates multi-resolution models, from which irregular or semi-regular triangulations are extracted to render a terrain at a suitable level of detail. Recent terrain simplification techniques use GPU-friendly regular grids and apply the filtering and sub-sampling paradigm to generate multiple resolutions. However, such approximations only sparsely adapt the terrain surface due to the smoothing and uniform sampling. Consequently, considerably more triangles have to be rendered in order to guarantee a certain error threshold.

In this paper we present a novel feature-sensitive simplification technique. Our approach follows the aforementioned paradigm. The key idea is to maintain the regularity and, at the same time, to recompute the vertex positions regarding a specific metric, the quadric error metric (QEM). Compared to previous methods we apply the paradigm to the grid of vertex-associated quadrics, from which the vertices for the new resolution are extracted by quadric error minimization. The benefit is, maintaining the regularity while taking the terrain features into account. As an overall consequence, the approximation error is decreased compared to methods based solely on vertex-filtering and sub-sampling. Hence, our method requires fewer triangles resulting in an improved rendering performance as reported in our results.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

---

## 1. Introduction

Terrain rendering has a long research history and is used in many application domains. Due to the huge amount of data, rendering systems make use of multi-resolution models to extract a specific *level of detail* to display. Starting from the most detailed representation, a multi-resolution model is generated a-priori by successively decimating the number of primitives with regards to a particular metric also known as *simplification*. The goal is to find a well-approximated model with less primitives. Traditionally, terrain multi-resolution models use irregular or semi-regular data structures at the granularity of individual triangles. However, such structures are complex and computationally expensive both for simplification and rendering. Hence, the recent multi-resolution model take advantage of triangle clusters (patches) [PG07]. Especially regular triangulations and data-layouts become attractive. Due to their simple layout and topology (valence

6) they are ideally suited for hardware processing [GGH02]. Furthermore, regularity guarantees efficiency in terms of memory management, serialization and rendering. Moreover, well-studied image processing and compression methods can be directly applied. Consequently, recent terrain simplification methods apply low-pass filtering - a weighted average filtering to avoid aliasing - followed by uniform sub-sampling to generate different resolution levels. In the following this process is referred to *filtering & sub-sampling* paradigm. However, such approximations sparsely adapt the terrain surface. This is due to the smoothing and the uniform sampling leading to higher approximation errors compared to traditional simplification methods. Consequently rendering effort increases with regards to triangles per error. The major problem addressed by this paper are the following:

- *Regular* layouts are well-suited for hardware processing

but the simplification mostly based on low-pass filtering and thus terrain features are “smooth away”.

- Complex algorithms use *irregular* data structures and recompute the vertex positions in such a way, that the approximation error is minimized. As a result the approximations are highly accurate but the computation is expensive and unsuitable for hardware processing.

The key idea behind our novel simplification approach is to recompute the vertex positions with regards to a error metric to minimize the approximation error while simultaneously maintain the regularity without a remeshing. This leads to the following questions: first how to recompute vertex positions to keep the error low in a efficient manner and second how to preserve the regularity? To accomplish this, we apply the filtering & sub-sampling paradigm in combination with the *quadric error metric* [GH97]. We initially define associated plane-sets - express as quadrics - for all vertices whereby the planes represents the terrain surface. By the mean of filtering we weighted combine the quadrics. For the new resolution resulting from sub-sampling, we compute the representative vertices by minimizing the quadric error. As a result, the approximation error is lower in contrast to vertex-based filtering and sub-sampling. Consequently less triangles per error needs to be rendered increasing the performance as shown in the results.

The paper is outlined as follows: in the next section we present the relevant related work. In Section 3 we introduce our simplification method and show how the QEM can efficiently use to recompute vertex positions. After that, we discuss in Section 4 how a multi-resolution model can constructed with our simplification algorithm. In Section 5 we discuss the results and finish in Section 6 with a conclusion and future work.

## 2. Related Work

A terrain rendering algorithms is mostly divide into two major steps: first, the preprocessing step which generates a multi-resolution model using mesh simplification methods and second the rendering step. There exists a wide range of multi-resolution data structures as well as simplification algorithms in the literature. In the following we focus on those applied to terrain rendering in particular.

**Multi-resolution models** has a long history in terrain rendering due to the desire of displaying large datasets. Traditionally *vertex hierarchies* [XV96, Hop98, HSH09] or *multi-triangulation hierarchies* [Pup98, CGG\*05] are a suitable choice from which highly adaptive and high quality irregular triangulations can be extracted. However, those general hierarchies have not been established in the field of terrain rendering, due to the complex nature and the expensive computational costs. Instead, sub-division schemes has developed, which produces semi-regular triangulations by recursive applying the scheme. Commonly *longest-edge-bisection* sub-division is a very popular scheme (see

[LP01]). This schemes is as simple as powerful and have been applied in various forms and data structures for instance [EKT01, DWS\*97, RHS98, LKR\*96]. Another popular hierarchy commonly used is the *quad-tree*. A quad-tree is build by hierarchical sub-divide the terrain in a restricted [Paj98, And07] or non-restricted manner [Ulr02].

With the evolution of GPUs, it becomes necessary to move the granularity from individual triangles to *patches* (triangle clusters). In more detail: instead associating a triangles or vertex with a node in the hierarchy, off-line precomputed and optimized triangle patches are used. Recent GPU-oriented multi-resolution hierarchies use both regularly triangulated patches and regular data-layouts which leads to new simplification methods. For the interested reader we refer to [Paj02, LP02, dFKP05, PG07, DGY07].

**Simplification** generates a approximation by controlled decimating the number of primitives. The goal is to find a approximation which rarely differ from the original. For this purpose determining the elements to remove is necessary. This process is more or less computational expensive and directly depends on the complexity of the data-structure. Irregular data structures allow complex operations and error metric. For instance *vertex-clustering* [RB93], *iterative edge contraction* [Hop96], *mesh optimization* [HDD\*93] or *wavelet analysis* [GGS95] techniques can be use to keep the approximation error low. For a good overview we refer to [CMS98]. Especially [CGG\*03] takes advantages of quadric-based simplification [GH97] to generate high quality patch triangulations. this class of algorithms generate highly accurate approximation. However, the computation is very expensive and not optimized for recent hardware processing.

In contrast to that, algorithms based on semi-regular data structures merge elements based on the recursive sub-division scheme and measure the introduced error. For instance [Pom00, Ulr02, SW06, DSW09, BGP09] generated semi-regular triangulated patches in an off-line process, whereas [LP01, Lev02] extracts the approximation during the rendering processes. In general such algorithms are very fast and easy to implement, due to the recursive nature. However, irregular triangulations leads to better results with regards to the triangles per error [EKT01].

For regular data structures methods from the image processing community has been adapted due to the close interrelation of  $n \times m$  grids of height values and images. Recent terrain simplification algorithms carried out the *filtering & sub-sampling* paradigm to generate multiply resolutions. In more detail: A approximation is generate by uniformly sub-sample the original regular input domain. To avoid aliasing artifacts, a weighted average (low-pass) filtering is applied [dB00, LH04]. [GMC\*06, BGMP07] use wavelets analysis, whereas [HDJ04] apply a high quality low-pass filter for triangular patches. These algorithms are quit efficient but wasting triangles per error due to a sparsely adaption of the terrain surface in contrast to irregular triangulations.

**Quadric Error Metric** is widely used in simplification processes, due to its efficiency and quality issues. The metric drives the simplification process by recompute vertex positions in such a ways that the approximation error is minimized. The metric has been developed for the *pair collapse* operator which merge two vertices connected by a edge. For each vertex  $\mathbf{v}$  the metric defines the point-plane distance from  $\mathbf{v}$  to a set of associated planes  $planes(\mathbf{v})$ .

$$f(x) = \sum_{\mathbf{p} \in planes(\mathbf{v})} (\mathbf{p}^T \mathbf{v})^2 \quad (1)$$

$$= \mathbf{v}^T \left( \sum_{\mathbf{p} \in planes(\mathbf{v})} \mathbf{p}^T \mathbf{p} \right) \mathbf{v} \quad (2)$$

$$= \mathbf{v}^T Q_{\mathbf{v}} \mathbf{v} \quad (3)$$

Initially, the plane set  $planes(\mathbf{v})$  is given by the adjacent faces of  $\mathbf{v}$ . If a pair collapse  $(\mathbf{v}_1, \mathbf{v}_2) \rightarrow \bar{\mathbf{v}}$  is applied, the position of  $\bar{\mathbf{v}}$  is compute by minimize the distance to the combined associated planes  $planes(\mathbf{v}_1) \cup planes(\mathbf{v}_2) \rightarrow planes(\bar{\mathbf{v}})$ . The explicit representation can be replaced by the *error quadric* a symmetric  $4 \times 4$  matrix  $Q_{\mathbf{v}}$  and the set-union operator reduces to quadric addition. It also possible to scale the contribution of a quadric by quadric-scalar multiplication. Due to the fact, that vertex-clustering is equivalent to performing the pair collapse operation to each vertex in a cluster simultaneously, the quadric error metric can be used as a measure of mesh quality. Furthermore it is possible to compute the representative vertex of a cluster that minimizes the error quadric. Based on this idea [] developed a real-time GPU based mesh simplification method which stores the quadrics in texture memory and computes cluster representants directly on the graphics hardware.

**Implications** for a simplification algorithm are the following: regular data-layouts should be use to exploit recent hardware, but the commonly applied low-pass filtering in the simplification leads to a wasting number of triangles/error in the rendering process. More complex techniques, iteratively recompute vertex positions with regards to a error metric, leading to high quality approximations. However such algorithms are complex and time consuming. Furthermore, the data structures and layouts are unfavorable for recent hardware. Hence, it should be expect that the combination of both will be beneficial as we confirmed with our approach proposed here.

### 3. QEM-Filtering

#### 3.1. General Procedure

1. vertex map generation (hf to vertices)
2. derive fundamental quadric map FQM
3. filter and sub-sample FQM
4. recompute vertex positions from FQM and evaluate

5. optional smoothing
6. successively

#### 3.2. FQM

1. high-field to vertices -> projection ?
2. connectivity (quads, tris)
3. tagential quadrics -> impact

#### 3.3. Filtering

1. filtering in spacial domain
2. adaption of QEM
3. filter weights and size -> impact

#### 3.4. Sub-Sampling

1. computation of the vertex position
2. optimization and evaluation of positions

#### 3.5. Discussion

#### 4. Application to Terrain Rendering

1. apply to data structures
2. patch-edges
3. smoothing

How to adapt the filtering subsampling to generate patches for quad-tree and bin-trees. Different sampling schemes.

#### 5. Results

1. Rendering system
2. comparison of different filter weights
3. comparison of rendering time 1D-displacement/3D-displacement
4. result discussion (time, resources, quality)

#### 6. Conclusion

#### References

- [And07] ANDERSSON J.: Terrain rendering in frostbite using procedural shader splatting. In *SIGGRAPH '07: ACM SIGGRAPH 2007 courses* (New York, NY, USA, 2007), ACM, pp. 38–58. 2
- [BGMP07] BETTIO F., GOBBETTI E., MARTON F., PINTORE G.: High-quality networked terrain rendering from compressed bitstreams. In *Proceedings of the twelfth international conference on 3D web technology* (2007), ACM, p. 44. 2
- [BGP09] BÖSCH J., GOSWAMI P., PAJAROLA R.: RASTeR: Simple and Efficient Terrain Rendering on the GPU. In *Eurographics 2009 - Areas Papers* (2009), Eurographics Association, pp. 35–42. 2

- [CGG\*03] CIGNONI P., GANOVELLI F., GOBBETTI E., MARTON F., PONCHIO F., SCOPIGNO R.: BDAM - batched dynamic adaptive meshes for high performance terrain visualization. *Computer Graphics Forum* 22 (2003), 505–514. 2
- [CGG\*05] CIGNONI P., GANOVELLI F., GOBBETTI E., MARTON F., PONCHIO F., SCOPIGNO R.: Batched multi triangulation. *IEEE Visualization, 2005. VIS 05* (2005), 207–214. 2
- [CMS98] CIGNONI P., MONTANI C., SCOPIGNO R.: A comparison of mesh simplification algorithms. *Computers & Graphics* 22, 1 (1998), 37–54. 2
- [dB00] DE BOER W. H.: Fast terrain rendering using geometrical mipmapping, 2000. 2
- [dFKP05] DE FLORIANI L., KOBBELT L., PUPPO E.: *Mathematics and Visualization*. Springer Berlin Heidelberg, Berlin Heidelberg, 2005, ch. A Survey on Data Structures for Level-of-Detail Models, pp. 49–74. 2
- [DGY07] DIETRICH A., GOBBETTI E., YOON S.: Massive-Model Rendering Techniques. *IEEE Computer Graphics and Applications* (2007), 20–34. 2
- [DSW09] DICK C., SCHNEIDER J., WESTERMANN R.: Efficient geometry compression for GPU-based decoding in realtime terrain rendering. *Computer Graphics Forum* 28, 1 (2009), 67–83. 2
- [DWS\*97] DUCHAINEAU M. A., WOLINSKY M., SIGETI D. E., MILLER M. C., ALDRICH C., MINEEV-WEINSTEIN M. B.: Roaming terrain: real-time optimally adapting meshes. In *IEEE Visualization* (1997), pp. 81–88. 2
- [EKT01] EVANS W., KIRKPATRICK D., TOWNSEND G.: Right-triangulated irregular networks. *Algorithmica* 30, 2 (2001), 264–286. 2
- [GGH02] GU X., GORTLER S., HOPPE H.: Geometry images. *ACM Transactions on Graphics* 21, 3 (2002), 355–361. 1
- [GGS95] GROSS M., GATTI R., STAADT O.: Fast multiresolution surface meshing. In *IEEE Conference on Visualization, 1995. Visualization'95. Proceedings.* (1995), pp. 135–142. 2
- [GH97] GARLAND M., HECKBERT P.: Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (1997), ACM Press/Addison-Wesley Publishing Co., p. 216. 2
- [GMC\*06] GOBBETTI E., MARTON F., CIGNONI P., DI BENEDETTO M., GANOVELLI F.: C-bdam - compressed batched dynamic adaptive meshes for terrain rendering. *Computer Graphics Forum* 25, 3 (sep 2006). To appear in Eurographics 2006 conference proceedings. 2
- [HDD\*93] HOPPE H., DE ROSE T., DUCHAMP T., McDONALD J., STUETZLE W.: Mesh optimization. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques* (1993), ACM, p. 26. 2
- [HDJ04] HWA L. M., DUCHAINEAU M. A., JOY K. I.: Adaptive 4-8 texture hierarchies. In *VIS '04: Proceedings of the conference on Visualization '04* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 219–226. 2
- [Hop96] HOPPE H.: Progressive meshes. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1996), ACM, pp. 99–108. 2
- [Hop98] HOPPE H.: Smooth view-dependent level-of-detail control and its application to terrain rendering. In *VIS '98: Proceedings of the conference on Visualization '98* (Los Alamitos, CA, USA, 1998), IEEE Computer Society Press, pp. 35–42. 2
- [HSH09] HU L., SANDER P., HOPPE H.: Parallel view-dependent refinement of progressive meshes. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games* (2009), ACM New York, NY, USA, pp. 169–176. 2
- [Lev02] LEVENBERG J.: Fast view-dependent level-of-detail rendering using cached geometry. In *VIS '02: Proceedings of the conference on Visualization '02* (Washington, DC, USA, 2002), IEEE Computer Society, pp. 259–266. 2
- [LH04] LOSASSO F., HOPPE H.: Geometry clipmaps: terrain rendering using nested regular grids. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers* (New York, NY, USA, 2004), ACM, pp. 769–776. 2
- [LKR\*96] LINDSTROM P., KOLLER D., RIBARSKY W., HODGES L. F., FAUST N., TURNER G. A.: Real-time, continuous level of detail rendering of height fields. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1996), ACM, pp. 109–118. 2
- [LP01] LINDSTROM P., PASCUCCI V.: Visualization of large terrains made easy. In *Proceedings of the conference on Visualization '01* (2001), IEEE Computer Society, p. 371. 2
- [LP02] LINDSTROM P., PASCUCCI V.: Terrain simplification simplified: A general framework for view-dependent out-of-core visualization. *IEEE Transactions on Visualization and Computer Graphics* 8, 3 (2002), 239–254. 2
- [Paj98] PAJAROLA R.: Large scale terrain visualization using the restricted quadtree triangulation. *Visualization Conference, IEEE 0* (1998), 19. 2
- [Paj02] PAJAROLA R.: Overview of quadtree-based terrain triangulation and visualization. *UCI-ICS Technical Report* (2002). 2
- [PG07] PAJAROLA R., GOBBETTI E.: Survey of semi-regular multiresolution models for interactive terrain ren-

- dering. *The Visual Computer* 23, 8 (2007), 583–605. 1, 2
- [Pom00] POMERANZ A.: *ROAM Using Surface Triangle Clusters (RUSTiC)*. PhD thesis, UNIVERSITY OF CALIFORNIA, 2000. 2
- [Pup98] PUPPO E.: Variable resolution triangulations. *Computational Geometry: Theory and Applications* 11, 3-4 (1998), 219–238. 2
- [RB93] ROSSIGNAC J., BORREL P.: Multi-resolution 3D approximations for rendering complex scenes. *Modeling in Computer Graphics* 455 (1993), 466. 2
- [RHS98] RÖTTGER S., HEIDRICH W., SEIDEL H.-P.: Real-time generation of continuous levels of detail for height fields. pp. 315–322. 2
- [SW06] SCHNEIDER J., WESTERMANN R.: GPU-friendly high-quality terrain rendering. *Journal of WSCG* 14, 1-3 (2006), 49–56. 2
- [Ulr02] ULRICH T.: Chunked lod, 2002. <http://www.tulrich.com/geekstuff/chunklod.html>. 2
- [XV96] XIA J. C., VARSHNEY A.: Dynamic view-dependent simplification for polygonal models. In *VIS '96: Proceedings of the 7th conference on Visualization '96* (Los Alamitos, CA, USA, 1996), IEEE Computer Society Press, pp. 327–ff. 2