

# Interactive Visualization of Large ISO-Surfaces

スヴェン フォストマン†  
Sven Forstmann

大谷 淳‡  
Jun Ohya

## Abstract

Presented is a novel algorithm for the real-time view-dependent visualization of large iso-surfaces. The developed method utilizes a unique continuous level-of-detail approach based on clip-boxes for visualizing the triangular geometry which is generated from the original iso-surface. The algorithm is especially focused on simplicity and performance for highly interactive and continuous frame rates.

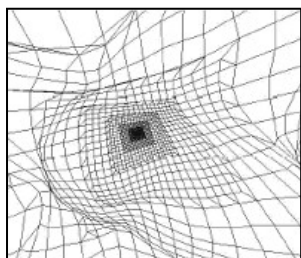


Fig. 1: Geometry by applying a height-map

## 1. Introduction

View dependent rendering of large Iso-Surfaces is a rarely researched topic in the area of visualization, but there have been several remarkable approaches, which will be introduced briefly.

One way to achieve continuous level-of-detail (LOD) has been the method of B. Gregorski<sup>1)</sup> to apply the well known ROAM algorithm, usually used for terrain rendering, in a 3D-manner for iso-surfaces.

A different idea by C.Erikson<sup>4)</sup> is to preprocess the whole iso-surface to generate HLODs with static LOD from the initial surface. The last and one of the most advanced methods is the algorithm of P.Lindstorm<sup>2)</sup>, by using vertex clustering.

## 2. The Algorithm

The here presented algorithm is inspired by the terrain rendering method of Losasso<sup>3)</sup>. Their approach uses geometry clipmaps for creating terrain geometry from height-maps, like in Fig.1, rather than building a complex LOD hierarchy.

The same idea has been used here, but redesigned for the three-dimensional case to visualize iso-surfaces from volumetric datasets. The clip-maps were therefore extended to clip-boxes, shown in Fig.2 (wire-frame) and Fig.3 (abstract).

While moving through the scene, the clip-boxes' positions are updated frequently to match the position of the visitor. The most inner box needs to be updated most often. The outer ones have to be updated in lower frequencies, depending on their size.

Almost all advantages listed in the Losassos<sup>3)</sup> paper for the two-dimensional case are preserved for the presented, three-dimensional case:

- *Simplicity.* It is not necessary to manage any complex tree structure and also time consuming preprocessing is unnecessary.
- *Optimal rendering throughput.* All geometry contained by a clip-box resides in Graphic-card memory in for of triangle-strips for optimal performance.
- *Steady rendering.* The frame-rate will not change abruptly by moving through the iso-surface, as there are no sudden changes in the tessellation.
- *Immediate complexity adjustment.* Even if the borders of the clip-boxes are fixed, it is possible to adjust them on the fly for reducing the amount of triangles.
- *Elegant degradation.* If the visitor is moving quickly, it might be impossible to update all clip-boxes fast enough. In such a case, starting from the largest clip-box, as many clip-boxes as possible are updated; note that higher levels of detail might be skipped for matching the desired frame-rate.
- *Memory usage.* As there is no additional data for tree-structures or error-values required, the whole system remains very memory friendly.
- *Synthesis.* The simple structure which is generated by the algorithm also allows an enhancement of the generated geometry with procedural details. In Fig.5 can be seen an artificial landscape where perlin noise has been applied.

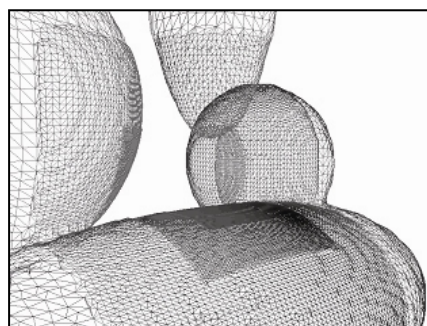


Fig. 2: A test scene generated by CSG in 3D

## 3. Results

The results of the presented method are shown in Fig.4, 5 and 6, where various artificial data-sets were tested. The size of the visible volume during testing was set to  $512^3$ , represented by 4 clip-boxes ( $64^3$  each). The triangle-count ranged from 50k for

† svenforstmann@yahoo.co.jp

‡ ohya@waseda.jp

low- and 400k for high-detailed scenes. The achieved speed can be seen in Fig.4 and was at an average of 100 Fps.

As the diagram certifies is the frame-rate also relatively continuous, one of the main design goals of this algorithm. By closer investigation, we can see that the consumed time per frame varies to a certain amount, caused by geometry updates. However, this fluctuation remains at background level and is hardly recognizable during interactive exploration.

Testing environment has been a 2Ghz AMD PC with Radeon 9800 graphics card.

While benchmarking, texture and background rendering have been switched off for optimal performance. The update-rate for the clip-boxes has been about 5Hz which was adequate for a smooth movement. The memory-consumption has been at a maximum of about 50MB.

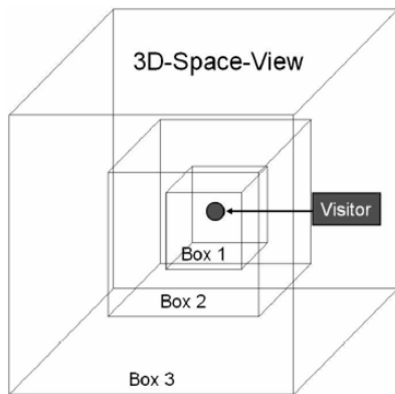


Fig. 3 Abstract 3D view showing 3 clip-boxes

#### 4 . Conclusion and Future Work

We have presented a novel approach for view-dependent visualizing of large iso-surfaces at highly interactive frame-rates, based on clip-boxes which have proven their effectiveness already the 2D case as clip-maps.

Future work for the presented algorithm might include generating tests on real-world volume-data-sets and also optimizing the smoothing method for improving the visual accuracy.

#### References

- 1).B.Gregorski: *Interactive View-Dependent Rendering of Large IsoSurfaces*, Visualization 2002.
- 2).P.Lindstrom: *Out-of-Core Construction and Visualization of Multiresolution Surfaces*, Siggraph 2003.
- 3).F.Losasso, H.Hoppe: *Geometry Clipmaps: Terrain Rendering Using Nested Regular Grids*, Siggraph 2004
- 4).C.Erikson, C.Manocha: *HLODs for Faster Display of Large Static and Dynamic Environments*, SigGraph 2001.

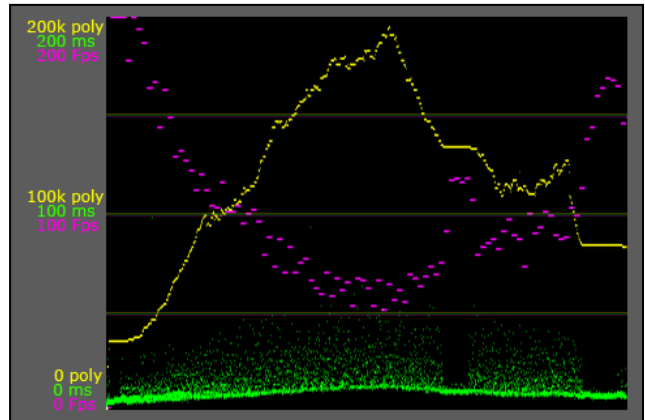


Fig.4: This diagram shows the benchmark results by flying through an example volume for 100 seconds. Green indicates the rendering time per frame, yellow the amount of polygons and purple the achieved frame-rate.



Fig.5: Artificial terrain

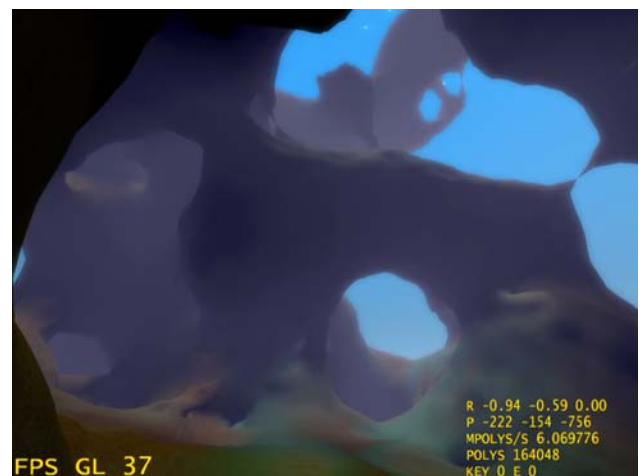


Fig. 6: Artificial cave example