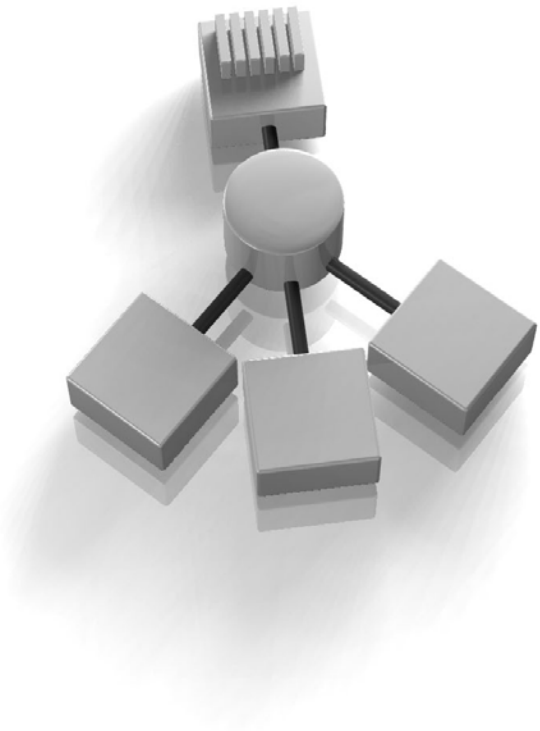
SIXTH FRAMEWORK PROGRAMME
PRIORITY IST-2002-2.3.1.8
Networked Audiovisual Systems

Uni-verse project
Deliverable D 6.1
Progressive reduction library
Apr 29, 2005



STREP project
Project acronym: Uni-Verse
Project full title: A Distributed Interactive Audio-Visual Virtual Reality System
Proposal/Contract no.: 002228
Distribution: Public

1 WP 6: "Scalability"

1.1 WP 6.1: Data Reduction

1.1.1 Introduction

In work package 6 it is proposed to make a geometry reduction library available for the rest of the implemented functionality in the project. As described in the final specification (see D2.3) this reduction library is based on a progressive simplification functionality that was already developed before the start of the Uni-Verse project within another project in the department A3 at IGD. It uses the progressive reduction algorithm described which was implemented in [1] and [2] to save effort and avoid redeveloping of already known technologies. The main tasks which had to be done within this work package were to adjust the existing reduction core to the special needs of the Uni-Verse project and to implement the corresponding extensions and requirements. The main goal was to cut out the reduction code from the already completed project and write a framework around that functionality to make it accessible. The material description had to be adjusted to allow the implementation of a different material specification according to the recommendations of the audio framework in work package 7 in the future.

The progressive reduction reduces the triangles in meshes based on an error metric for visual purposes. The process of removing triangles and the information on the triangles is saved in a list to be able to rebuild the original mesh from that information. This is standard approach like it is realized in other progressive compression systems as well.

The resulting library can now be used e.g. by rendering clients for PDAs and other slim devices with low processor power. They will be able to access a pre-compressed version of the currently hold geometry world of the server. Because this reduction is progressive the client can order a level of quality according to its performance possibilities. Another field of application is available in work package 7: The reduction library will be used for geometry reduction needed for audio calculations in that work package. There, it is not necessary to compute the complete detailed 3d world. For the most cases it is sufficient to use very simple and reduced objects.

The reduction library was developed using C++. It has no dependencies on other libraries except the standard C++ libraries and can therefore easily be integrated in other applications.

1.1.2 Verse Binding

Because the reduction algorithm needs special topology information it is not usable to bind the simplification directly to the Verse protocol. This will happen later with the integration of the library in the backend of the Alternative Verse Server developed in WP 6.2. Including the library that way it will become accessible through the Verse

protocol. The clients will then use Verse method calls to communicate with the reduction functionality. The Verse geometry data on the Verse server will be converted internally into a format the library can use. There is the need of e.g. defining meshes, normals and relations between vertices and polygons that is not explicitly mentioned in the Verse data structure.

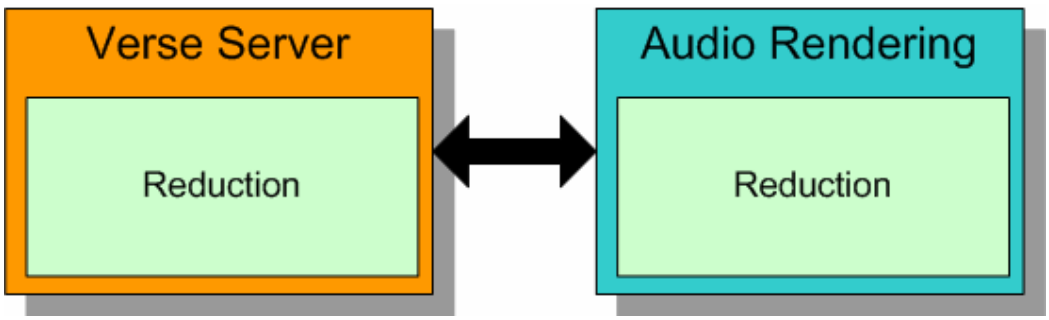
1.1.3 Error Metric

The error metric included is mainly vertex based. Every vertex gets a priority that is calculated using the surrounding geometric attributes. The decision which vertex will be removed or not later in the reduction process is based on these priorities. The exact description of the algorithm can be found in [1] and [2]. If the priorities are calculated, the vertices with high priority are removed using half edge collapse method. In every simplification step all vertices with low priority will be removed. The description of the removed vertices and their surrounding topology are saved in a progressive list. When the simplification is complete the execution of the list in reverse order will reconstruct the original geometry step by step.

1.1.4 Deliverable

The progressive simplification module is delivered as a static library. It is tested under Linux and Windows but no special system dependent libraries were included that could restrict the use on other different systems. It loads data from either the internally specified progressive file format or VRML files and includes all necessary commands needed for geometry reduction and its steering. The VRML file format was mainly integrated for fast testing purposes during the development. It is possible to export geometry data to the progressive file format or VRML. When saving compressed data to the VRML file all progressive information is lost and only the reduced 3d geometry will be written to the file.

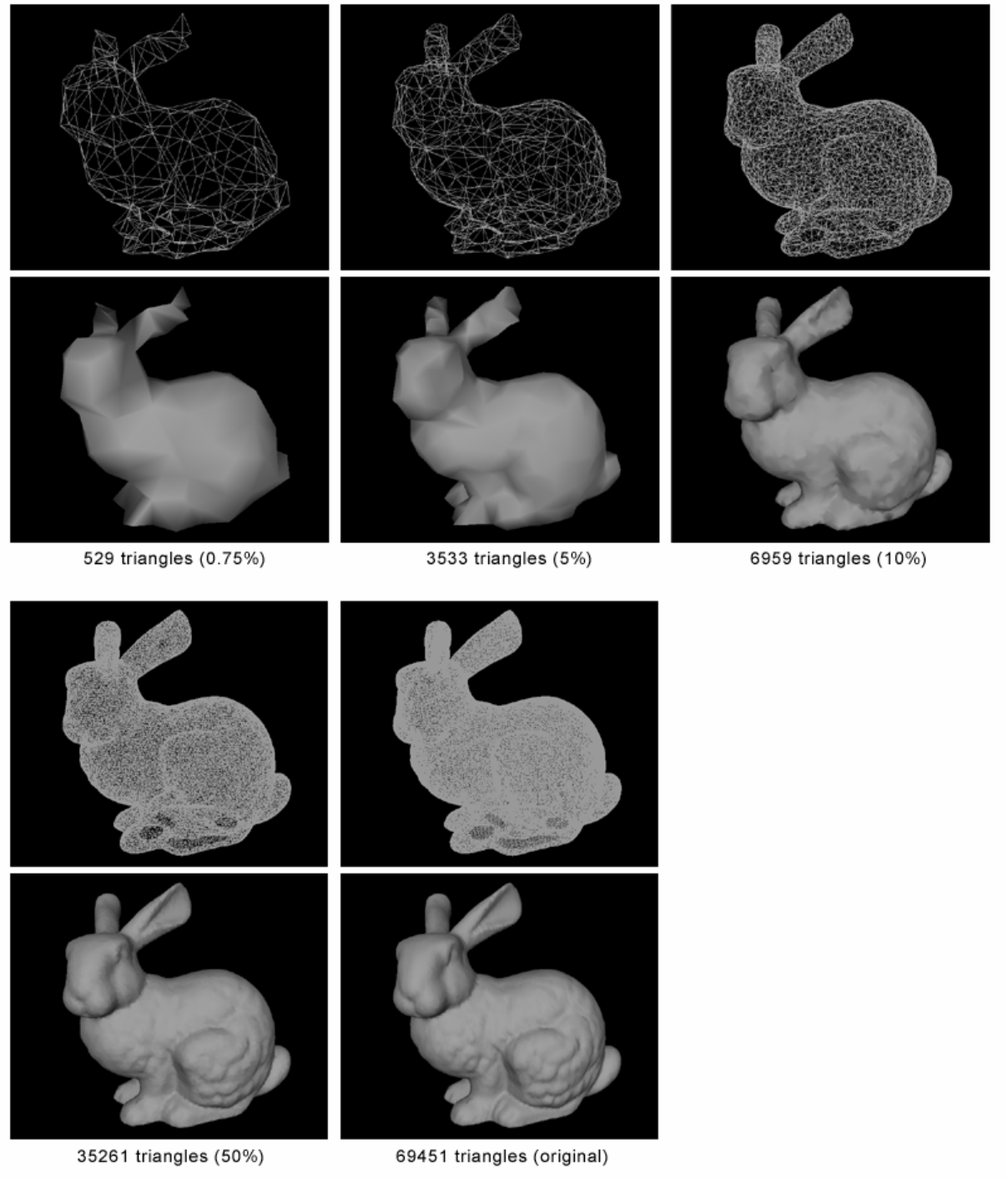
The reduction library core itself will not be available as open source because the main functionality originates from another project which was already completed at FIGD. The underlying licence is different from the open source general public licence.



Multiple use of the reduction library in the Uni-Verse project: Integration of the library into different Verse applications.

At this point in the project, the reduction library was already integrated and tested by the developers of work package 7. In the course of testing we came to the conclusion that this library is usable for the work of work package 7 but has to be extended in the near future. Material properties according to the audio recommendations must be adjusted and re-implemented. This will be done immediately after determining how the audio materials will look like. Furthermore, an advanced concept of geometry reduction with focus on audio calculations will be researched and developed in close cooperation with the leaders of work package 7.

2 Results



Reduction of the Stanford reference model bunny using the delivered progressive reduction library.

3 References

[1] J. Sahm, I. Soetebier, and H. Birthelmer. Efficient representation and streaming of 3d scenes. Computer & Graphics, 28(1):15–24, 2004.

[2] I. Soetebier, I. Birthelmer, J. Sahm, and V. Luckas. Managing large progressive meshes. Computer & Graphics, 28(5):691–701, 2004.