Automated conversion of CAD data in 3D low-poly models

Dennis Becker

Abstract— **An automated conversion of CAD data into 3D low-poly models would offer great potential in the industrial environment. Automated transformation will allow companies to continue to use CAD models, which are already in existence. The converted models can be used, for example, in the field of virtual reality or even on mobile devices for high-performance and graphically appealing visualizations. This work addresses the question of how and in what quality such an automatic transformation is possible. By means of a prototypical implementation it is shown that such an automated conversion is possible and which problems occur.**

I. Low-Poly-models

The topic and aim of this study are to investigate the extent to which existing CAD models can be fully automatically converted into high-performance and visually high-quality low-poly models. Low-poly-models are polygon (faces) meshes in 3D computer graphics, that has a relatively small number of polygons. However, as you can see in Figure 1, the optical quality of an object decreases significantly as the polygon number decreases. Round objects appear increasingly angular, many details are lost. In order to counteract this effect, numerous methods are used in game development to make low-poly models look high-quality. As many as possible of these methods should be used in the automatic conversion.
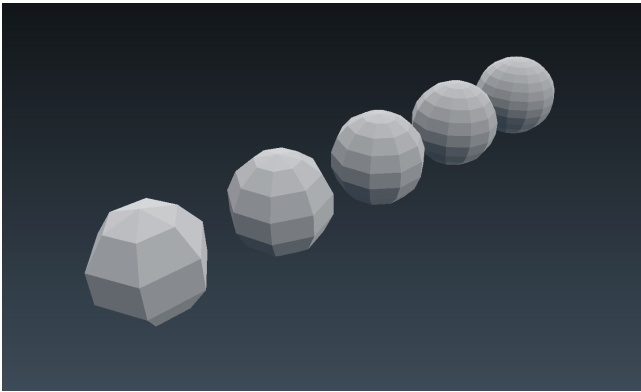

Fig. 1.   ..

The created low poly models could be used for high quality realtime rendering, for example in industrial virtual reality applications. But also in the area of simulations, configurators or in mobile applications, the low-poly models could be used.

The creation of low-poly models is state of the art in the game industry. The low number of polygons and the associated low computational complexity allows almost photorealistic scenes in computer games to be rendered in real time [1]. Here, however, the models are created manually by experienced artists. Although these models are accordingly very high quality, but also elaborate in their creation. It also requires extensive expertise for the manual creation of such models.

II. automated conversion

In contrast to this, software for the fully automatic conversion of CAD models could also be easily operated by laymen and would also be much faster. Thus, the automatic conversion could serve as the basis for high-performance 3D programs in the industrial and architectural sectors.

In order to examine this topic more closely, a concept was first worked out. The concept is based on the use of the open-source software Blender in combination with a specially developed Java control program.

To test the concept, a prototype implementation was made. The prototype can automatically convert polygon CAD models into low poly models. In the subsequent practical tests, the low-poly models not only had a significantly lower polygon count, but were also cleaned up for various possible sources of error. In this process, the metadata, for example the hierarchy of the CAD data, material settings and naming, can be adopted. In addition, the prototype shows some problems and limitations. Figures two and three show, by way of example, the CAD model of a turboshaft engine before and after the conversion. The corresponding parameters of the conversion can be found in Table 1. It shows that a polygon reduction of 90 % and even more is possible. Also, display errors could be removed at the bottom of the model.
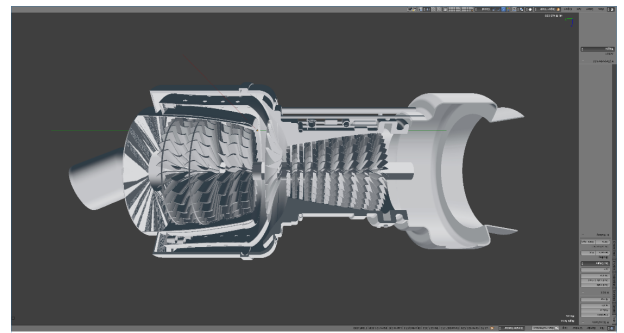
CAD-Modell: https://grabcad.com/library/turbo-shaft-helicopter-engine-1

Fig. 2.   Turboshaft engine, cad-modell 1

TABLE I

VERGLEICH POLYGON-MODELL UND LOW-POLY-MODELL

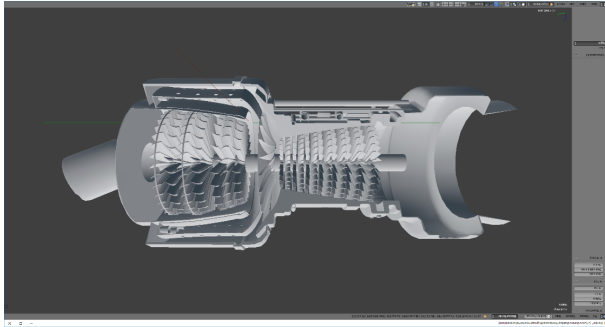|	Polygon-Model	Low-Poly-Model	Reduction
Polygons	988.129	92.803	90,6 %
Vertices	495.917	47.901	4,59 %
File size (.obj)	104 MB	5,48 MB	94,7 %

Fig. 3.   Turboshaft engine, low-poly-modell

III. CONCLUSION

Overall, it can be said that an fast and easy automatic conversion of CAD data into high-quality low-poly models is possible. But it also turns out that automating low-poly techniques from game development is a big challenge. The developed prototype could serve as the basis for a comprehensive implementation in the future.

REFERENCES

[1] C. Seifert, Spiele entwickeln mit unity 5, 2. auflage, München: Hanser Verlag.