Determination and Interactive Visualization of 3D-objects

Frank BOOCHS, Carsten GARNICA , Frank WOLTER
i3mainz, Institut für Raumbezogene Informations- und Messtechnik
Fachhochschule Mainz
Holzstr.36, D-55116 Mainz
e-mail: boochs@geoinform.fh-mainz.de

Commission V, WG V/3

KEY WORDS: Visualization, data structure, 3D-object, digital photogrammetry, quality attributes

ABSTRACT

The paper describes a procedure designed for the determination and visualization of surfaces from objects having full 3D-extent. The procedure is based on digital images captured from all sides around the object. The images will be triangulated and evaluated by means of image matching techniques, resulting in the generation of a large number of individual surface points densely scattered over the object surface. From this point field, the surface will be derived by triangulating all points into a TIN data structure. When the data structure is established, attribute information is incorporated, describing aspects being of interest for further usage of the data. Finally, the object surface will be visualized interactively, allowing for a user controlled inspection of the object. The visualization software is based on graphic standards and allows to display the object in 3D-perspectives which are generated from the object geometry, the visual information of the digital images and the coded presentation of the attribute values.

ZUSAMMENFASSUNG

In dem Aufsatz wird ein Verfahren zur Bestimmung und Visualisierung von räumlich geschlossenen Objekten vorgestellt. Die Bestimmung der Oberfläche beruht auf photogrammetrischen Techniken, mit deren Hilfe eine allseitige photographische Erfassung in Meßbildern sowie deren Triangulation und rechnergesteuerte Auswertung mittels eines Bildzuordnungsalgorithmus erreicht wird. Aus den aus der Auswertung hervorgehenden Punktfeldern wird die Oberfläche durch Dreiecksvermaschung der Objektpunkte und Aufbau einer TIN-Datenstruktur gewonnen. In die Datenstruktur werden neben den Koordinateninformationen auch Attributwerte aufgenommen, die geometrische, thematische oder visuelle Inhalte beschreiben können. Schließlich erfolgt eine interaktive Visualisierung der Oberfläche, die es erlaubt, alle Bereiche des Objektes räumlich zu inspizieren. Die Darstellung ist aufbauend auf Graphikstandards programmiert und gestattet neben der Darstellung der Geometrie des Objektes auch die Wiedergabe der Oberflächenradiometrie und der gespeicherten Attributwerte.

1 MOTIVATION

1.1 Introduction

Today, the power of computer assisted tools allows to model the environment or objects therein by means of algorithms and data sets describing the objects of interest. Possible applications range from simple documentation purposes to modeling tasks, simulations and support of construction processes, for example.

However, the increased use of discrete data sets will soon be extended to objects with high complexity. And although it is relatively simple to model artificial objects with the computer - even if they are represented in their full 3D-extent - it is very demanding to determine and model real 3D-objects due to their complex structure.

There are different techniques allowing to determine points in the 3D-space with the aim to establish point fields describing the surface of an object of interest. In practical use there are methods like:

- mechanical systems with tactile sensors generating precise coordinates even for objects with complex, non continuous shapes
- optical systems using coded light projected onto the object providing numerous precisely coordinated points on homogeneous surfaces
- laser or other range systems scanning the object surface and collecting a great number individually coordinated 3D-points
- photogrammetric techniques, allowing to determine the surface by automatic and/or interactive evaluation of metric images taken from the object in question.

All methods have their advantages and restrictions. Depending on the application it has to be decided which one seems to be the most promising (Heinz, 1998). Photogrammetric techniques are useful if flexibility in the data collection is required and if the reflectivity of the object has to be captured, too.

1.2 Requirements for 3D-objects with fully closed surface

In case of objects having a full extent in the 3D-space some further aspects have to be considered.

Optical and range systems observe the object from individual perspectives. This restricts the data collection to that part of the object facing the observing system. The determination of the full extent of the 3D-surface therefore needs several steps which have to be harmonized in order to generate a homogeneous description for the whole object. As complex surfaces show hidden parts if imaged from a single perspective, the setup of all views has to be selected carefully in order to allow the collection of the whole surface without gaps.

Of further importance is the question how to generate the surface from the individual point data collected. The points have to be combined into surface elements truly representing the object at that location. The data structure should be able to represent edges and more than one z-value with respect to the one xy-value.

Additionally, it should be considered that an efficient use of 3D-objects needs a visualization. This should not be restricted to the geometry only but also allow the representation of the object as it is in nature, requiring the capture and storage of visual information together with the geometry.

Finally, it might be of interest to keep several quality measures arising from the determination and aggregation of the point data. Accuracy values or geometrical attributes which have been evaluated at any step of the procedure are useful information for the final inspection of the surface and should be integrated into the surface data.

1.3 Available techniques

As the solution presented here is based on digital photogrammetry, algorithms and knowledge available may serve as base.

Mainly, the determination of the surface is achieved by a closed group of metric images arranged around the object, completely covering the surface. All images are triangulated in a common 3D-coordinate system. From image pairs forming stereo models the imaged surface parts are evaluated. The surface is expressed by dense point fields captured by means of image matching using the ARCOS program system (Boochs, 1989). As the orientation of the images is defined in the common 3D-coordinate system, the individual object points collected locally can be transformed into this system.

1.4 Required developments

The developments needed are concentrated on several aspects:

- design and realization of a data structure fully describing the closed object surface in the 3D-space
- conversion of the mass of individual points into an adequate surface description
- incorporation of surface reflection data into the structure
- incorporation of qualitative and geometrical attributes
- development of a tool for the visualization and interactive inspection of the object based on geometry, reflection and attributes.

In the following, these developments are outlined. Their explanation is divided into a part describing the data structure and the conversion of the object points and another one showing the visualization aspects and giving some examples.

2 THREE-DIMENSIONAL DATA STRUCTURE

2.1 Preprocessing

As already mentioned, the determination of the surface points is done by evaluation of several stereo models in their local coordinate system. These local models normally have an overlap to adjacent stereo models resulting in multiply determined surface parts in those areas.
Depending upon the extension of the object and the point density selected for the grids there may easily exist a field of more than a million individual points. In order to use such point data to generate the object model some preprocessing tasks have to be performed:

- in the overlapping model parts closely adjoining points are averaged in order to improve the data quality. Thus, the data quantity is reduced without loss of significant information
- points which are determined incorrectly and do not belong to the surface are detected and eliminated
- optionally, a simple filtering process may be executed, aimed at a reduction of noise effects.

2.2 Generation of topology

The preprocessing provides a large number of individual points located in a uniform 3D-coordinate system. This point field has to be used for the computation of the digital surface model. But individual points simply describe spots on the surface, necessitating a combination of adjacent ones to planar elements to describe the surface itself. This might be achieved in a good approximation by formation of three-dimensional triangles out of closely adjoining points and a subsequent combination of all triangles to a three-dimensionally closed surface. This strategy is suitable particularly if the surface cannot be described by geometrical objects (Constructive Solid Geometry) or by simple parametric surfaces.
Such a triangulation is usually performed using a Delaunay algorithm, what might be realized by one of the numerous programs available in that field. However, a simple application of this technique in the 3D-space is not optimal, because the object is thereby segmented into small volumes, finally representing the three-dimensional convex hull of the object.

2.3 Generation of the data structure

In order to allow the application of simple Delaunay algorithms, the surface is separated into individual parts. These parts have to be arranged in a way that a Z(X,Y) representation can be accepted. For these parts, the triangulation will result in a correct approximation of the surface hull. The generation of the whole surface is finally performed combining the results for all adjacent parts.

2.3.1 Surface partitioning

The surface has to be divided into sections as fine as necessary to allow an individual processing. It must be possible to connect the points by means of a two dimensional Delaunay triangulation to a triangle network (TIN) within each partion. To meet this condition, a local coordinate system is defined where the surface is described with a function Z_local = f (X_local, Y_local), which does not produce any ambiguities.
If the object surface is simply shaped or if it can be described systematically (e.g. if it is similar to a sphere), the subdivision of the surface might be done automatically. Often, the partitioning is similar to the arrangement of the photogrammetric stereo models. But the more complicated the shape of the surface is, the more surface sections are necessary, resulting in a corresponding effort. As an example see figure 1.

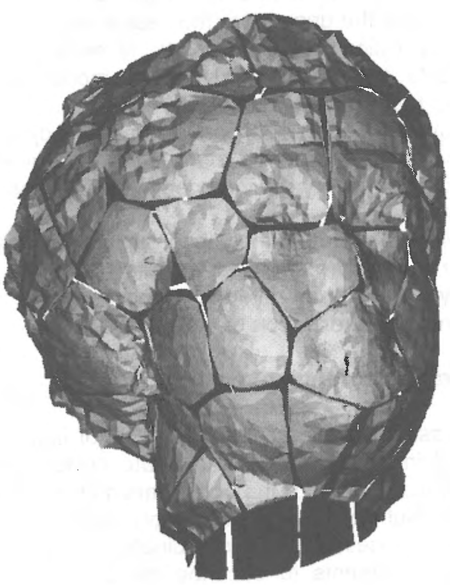


Figure 1: Division of a surface into individual processed parts

2.3.2 2D-Delaunay-Triangulation

For each defined surface part the points are transformed into the appropriate local coordinate system. In addition, border points from the adjacent sections lying close to the current one are transformed, too. This is of importance because the final combination of adjacent TIN's needs to know the points in the border zone.
Now an ordinary two dimensional triangulation is derived from the transformed points.

2.3.3 Combination of adjacent TIN's

The triangulation is successively processed for each individual surface part. When a new triangulation starts, the triangles already generated in the overlapping zone to adjacent sections are treated as constraining conditions (cf. fig. 4). This is necessary to ensure that the new TIN corresponds to the already existing ones. As a final result, a closed and not contradictory triangle network of the entire object surface will be established.

2.4 Requirements and restrictions

An object which is to be reconstructed by the procedure described above, has to fulfil only few requirements with respect to its shape. It may be unrestrictedly extended in all three axes of the 3D-space. The surface does not have to be convex, especially "overhangs" are not a problem. Objects that are only 2.5D (i.e. objects that can be completely described by a Z = f (X,Y) function) can also be processed by the procedure.

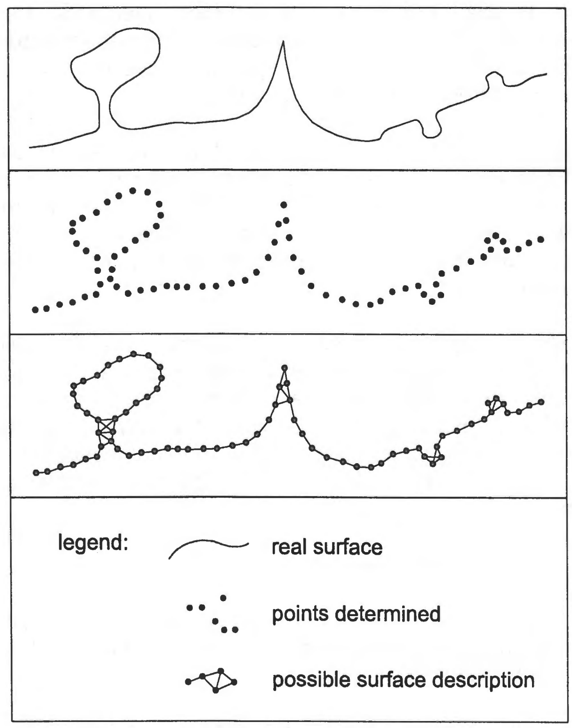


Figure 2: Correctness of the TIN in dependence on the surface shape

From the fact that the generation of the topology is based on the distance of closely adjoining points, a reconstruction is possible only if not adjacent surface parts don't lie closely together in the 3D-space (cf. fig.2).
The data set should not contain too many locally grouped blunders. In such a case the triangulation might assign these incorrect points to a non existent surface (cf. fig.3).
If these requirements are not met the procedure will not be able to reconstruct the surface uniquely.

2.5 Representation of the object in the data structure

The result of the procedure outlined above is a TIN-model of the object. This model represents the geometry and is combined with attribute information describing surface characteristics and quality. Together with the geometry, this attribute data is arranged in a special data structure.

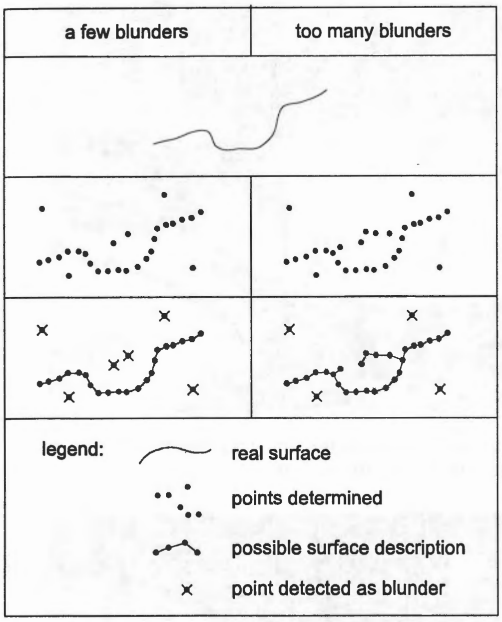


Figure 3: Correctness of the TIN in dependence on the point quality

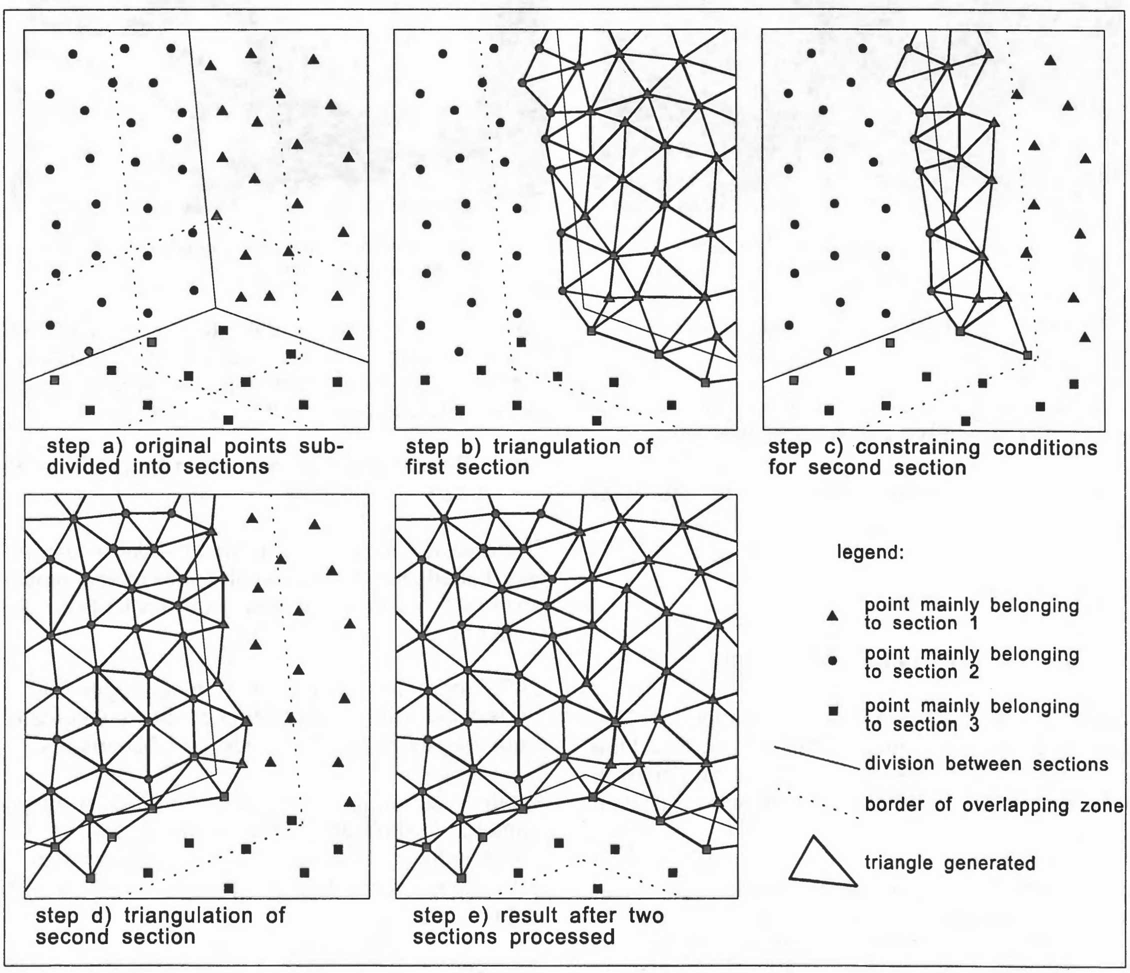


Figure 4: Combination of adjacent TIN's

The **geometry** is represented in the data structure by a point list and a triangle list. The point list contains the three dimensional coordinates of all surface points. The triangle list describes which three points form each triangle. All triangles together represent the entire surface of the object (cf. fig. 5a).

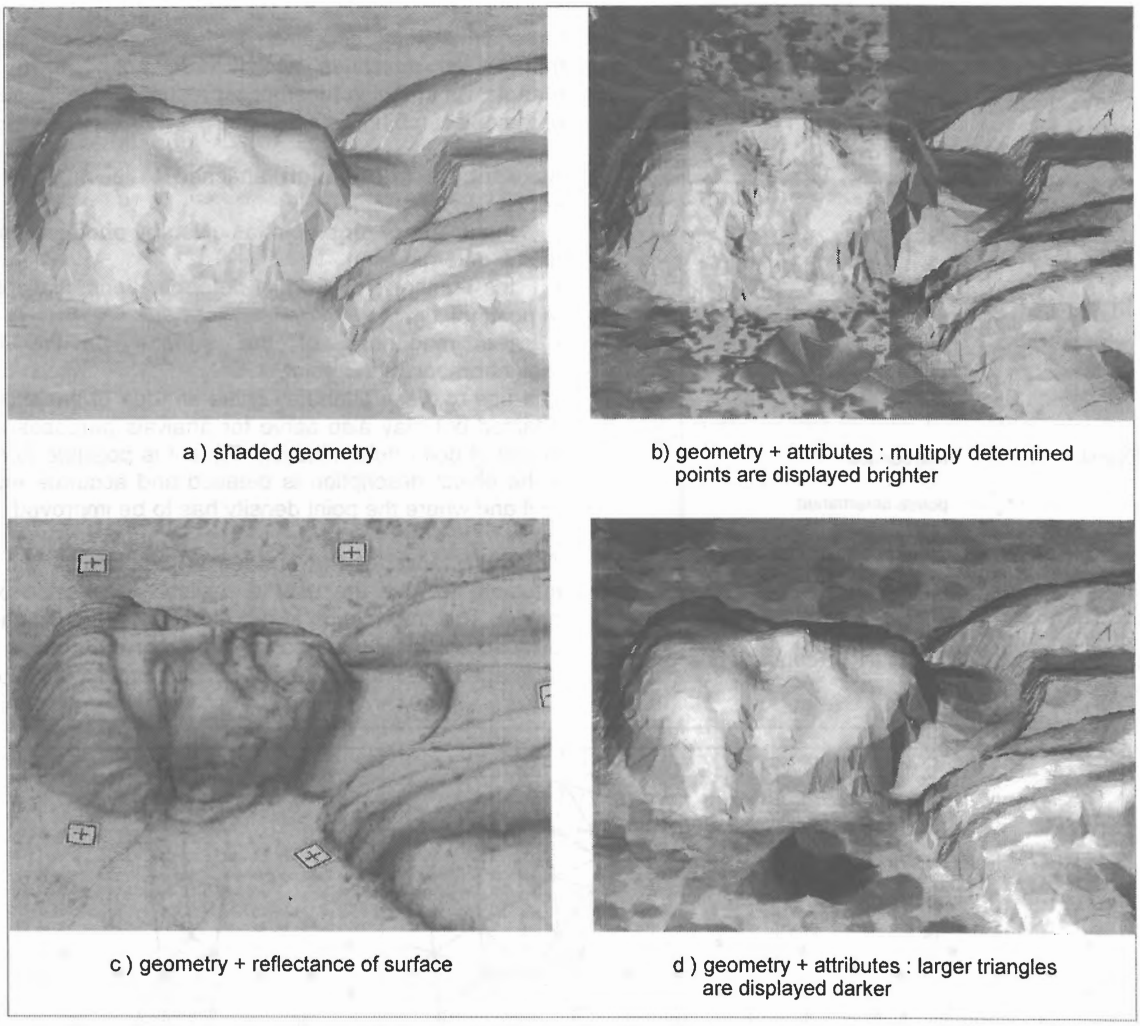
As **attribute information** attached to each point may serve:

- the accuracy for a point as given by photogrammetric point determination
- the number of individual determinations available for a point (cf. fig. 5b)
- the roughness of the surface in the direct neighborhood of the point.

The use of these attributes gives an idea of the accuracy attained but may also serve for analysis purposes in the phase of point determination. Thus it is possible to check if the object description is detailed and accurate enough or if and where the point density has to be improved.

The reflectance of the surface is included in the data structure to give the user a realistic impression of the object. This information is not stored directly, instead the pixel coordinates of all those pictures are marked, a point is imaged to. Thus, it is possible to project the surface reflectance information as texture onto the object surface (cf. fig. 5c).

Derived (implicit) attributes such as the size of the triangles (cf. fig. 5d) or the shape of the triangles (relation



a) shaded geometry

b) geometry + attributes : multiply determined points are displayed brighter

c) geometry + reflectance of surface

d) geometry + attributes : larger triangles are displayed darker

Figure 5: Different visualizations of an object

between longest edge and shortest height) can be calculated and integrated into the data structure.

Further attributes can easily be added to the data structure.

3 VISUALIZATION

3.1 Aims of the visualization

The visualization is performed by 3D-rendering. This includes the calculation of hidden-surface removal and the calculation of lighting with respect to shading and color. It is also possible to execute rotation, scaling and translation of the object.

The algorithm developed can either visualize points or triangles. The visualization of the triangles is used to observe the surface with radiometrical details, resulting in more time consuming computations. The visualization of points gives an impression of the model and can be interactively controlled by the operator in real time.

One major aim is the visualization of accuracy attributes. These attributes are visualized in different colors. So the observer can easily determine in which areas of the surface inaccuracies might be present.

3.2 Development of a custom-built visualization tool

The decision to develop a custom-built visualization tool was based on the individual performance requirements which are not met completely by available tools for 3D-rendering.

CAD-Programs, for example, mainly generate volume models as used in Constructive Solid Geometry. Such programs are not able to render real 3D-images.

Another alternative is the use of an abstract programming language to develop 3D-Rendering algorithms. Programs written in these languages sometimes turn out to be rather slow when visualizing a great amount of geometric primitives.

However, the Graphics Library OpenGL is simply a collection of rendering functions, so it can be customized to the needs of the visualization. Some versions of the OpenGL can access the graphics device directly allowing to increase the rendering performance considerably.

3.3 Required functionality

In the phase of the generation of the object model, a visualization tool can be used to detect inaccuracies and/or deficiencies of the model. After the model is computed correctly, the visualization can be used for presentation.

For interactive transformations of the model the performance of the rendering must be high and it must be possible to visualize a great amount of data. This is achieved using one viewport for the visualization of the surface with triangles and attributes and another viewport which simply renders points located on the surface of the model, leading to fast performance.

For a simple and intuitive operation of the program, a graphical user interface (GUI) has to be implemented. And for further customization to different displays, the GUI consists of different widgets, one for geometry, one for radiometry and one for quick visualization.

As an example for the realized GUI see figure 8.

The program is designed for a use on UNIX–workstations.

3.4 Realization

3.4.1 Components needed

The program can functionally be subdivided into three components:
Routines for data storage and processing, routines for 3D-rendering and a GUI. These components are realized using the Graphic Library OpenGL and the programming languages C and Tcl / Tk.

3.4.2 Components used

For effective memory administration the programming language C was chosen. It provides fast memory access and controllable allocation of memory.
OpenGL is mainly a Z-buffer and provides fast rendering. OpenGL serves most functions of a 3D-Renderer. As a library it can be customized to the needs of the program.
TK is an extension package to the interpreted programming language Tcl. It is an abstract tool to create GUIs and simply allows to design GUIs (in comparison to Motif window techniques or raw X-Programming). Nevertheless it is possible to create GUIs with a broad range of widgets for the purposes of the interaction of the user.

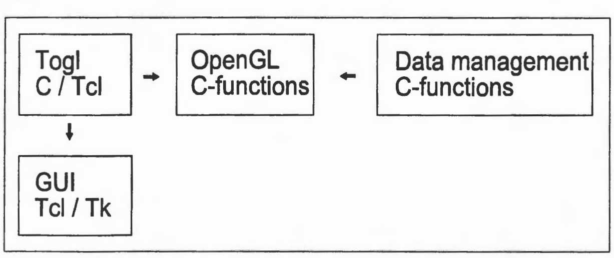


Figure 6: Components of the visualization program

To port an OpenGL- graphic into Tcl / Tk a special widget is needed. It has to have an image memory, color with a model (RGB vs. color index model) and a depthbuffer. Togl is a Tk-OpenGL-widget. Some of its functions can be called by C-functions, others with Tcl / Tk, so it is an interface between the GUI and the 3D-renderer (cf. fig. 6).

3.4.3 Linking the components

The components of the program have to be linked to an executable Tcl-Interpreter. This is possible because Tcl is written in C and is meant to be compiled, together with other routines in C, such as own functions, Tk or OpenGL.

By running the program the Tcl-Interpreter invokes the routines written in C. Therefore no loss of performance of the OpenGL and the data administration is caused, although Tcl is an interpreted language (cf. fig. 7).

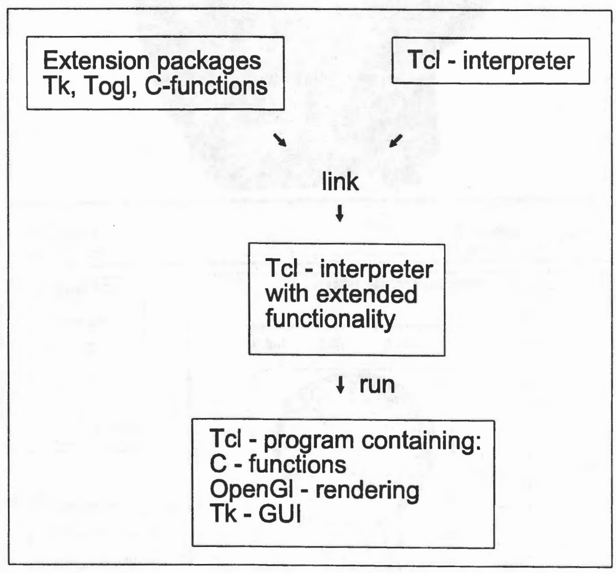


Figure 7: Linkage and execution of the visualization program

3.5 Restrictions and performance

Texture mapping depends on the number and extent of the digital pictures. With 26 pictures each with an extent of 4500 x 4500 pixels it takes 5 min. 20 sec. in pyramid level 1 respectively 25 sec. in pyramid level 4 including the time to read the image data from the hard-disk.
Rendering of the object model takes 5 sec. with 151.242 triangles, 3 sec. with 57.632 and 1 sec. with 9.310 triangles. OpenGL has to perform the texture mapping for each image and the rendering of the triangles, so the time values can simply be summarized.
The tests were performed on a SUN ULTRA 2 workstation with a Creator 3D-graphics device.

The configuration of the OpenGL can be changed directly. So it is possible to modify the performance directly by changing the lighting model, shading model and reflectance resulting in an corresponding change of the quality of the visualization. These settings can be changed very easily, as well as the integration of other attributes for the points or the triangles.

However, the compact structure of the program does not allow to incorporate completely new and complex features without a considerable effort.

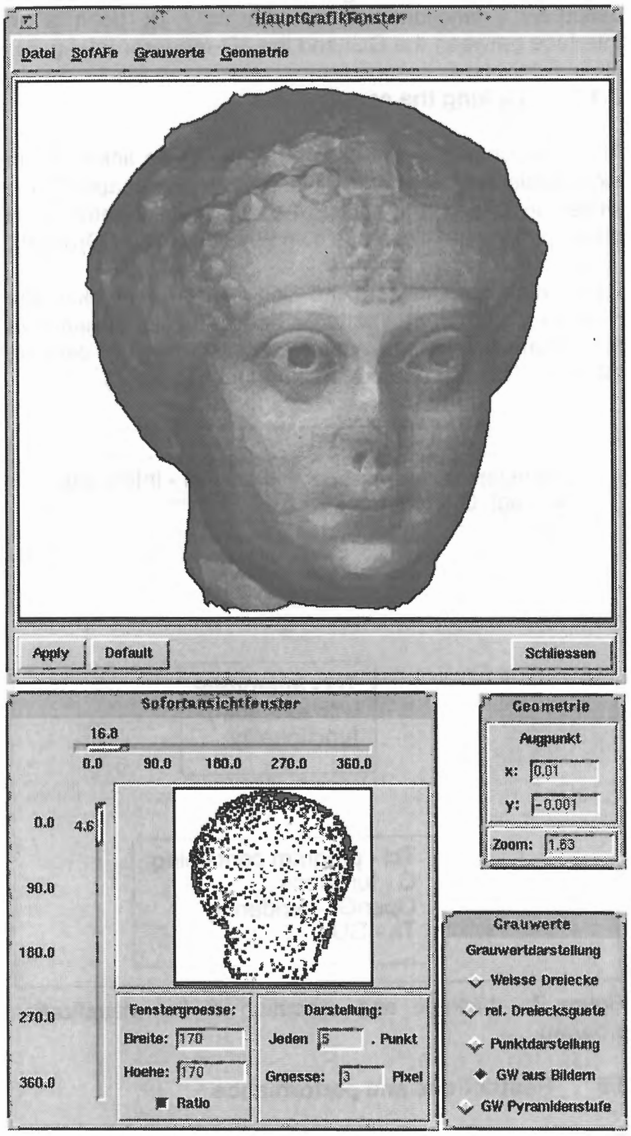


Figure 8: The GUI used for interactive control

4 SUMMARY

The article shows that it is possible to apply well known techniques in the field of photogrammetry, data processing and computer graphics to the demanding task of determination and visualization of 3D-objects. The procedure is applicable to all types of 3D-objects and gives not only the complete geometry but also useful quality and accuracy attributes together with the surface reflectance. This allows an interactive visualization, giving the opportunity to inspect the object from all sides, in all perspectives and together with the attributes collected.
Thus, the user is provided with a useful tool for the precise determination and visual analysis of 3D-objects.

5 ACKNOWLEDGEMENTS

The project has been funded by the federal ministry of research and technology under contract No. 1702596.

6 REFERENCES

Boochs, F., Hartfiel,P., 1989. Festigung von Sicherheit und Zuverlässigkeit der DHM-Bestimmung bei Einsatz der Bildkorrelation. Bildmessung und Luftbildwesen, Vol. 57, Nr. 3

Boochs F., Heinz G. 1996. Generation and Use of Digital Surface Models for Volume Objects. International Archives of Photogrammetry and Remote Sensing, Vol. 31, No 3.

Heinz G. 1998. Comparision of Different Methods for Sculpture Recording. International Archives of Photogrammetry and Remote Sensing, Vol. 32, No 5.

IGD, 1997, Sculptor – a 3D-scanning system for the reconstruction of object shapes. Fraunhofer Institut für graphische Datenverarbeitung.

Kürten A., 1998. Vergleich verschiedener Vefahren zur Erzeugung digitaler Objektmodelle. Diplom thesis, unpublished.

Neider, J.; Davies, T.; Woo, M.; 1993. OpenGL Programming Guide. Addison-Wesley.

OpenGL Architecture Review Board, 1992. OpenGL Reference Manual. Addison-Wesley

Ousterhout, J. K., 1995. Tcl und Tk. Addison-Wesley.

Paul, B., 1996. Togl – a Tk OpenGL widget. URL: http://www.ssec.wisc.edu/~brianp/Togl.html.

Riechmann W., Thielbeer B., 1997. Hochaufgelöste Oberflächenerfassung. Photogrammetrie, Fernerkundung, Geoinformatik, Heft 3, S. 155-164.

Shewchuk, J.R., 1997. Software for a 2D Delaunay Triangulation. URL: http://www.cs.cmu.edu/~quake/triangle.html.

Wahl F., 1984. A coded light approach for 3D-vision. IBM research report RZ 1452.