

Creating Morph Targets with an Active Triangulation Scanner

Tibor Kovács

Department of Automation
Budapest University of Technology and Economics,
H-1111 Budapest, Goldmann Gy. tér 3.
HUNGARY
E-mail: kovacs.tibor@aut.bme.hu

Creating morph targets in 3D environment is not the easiest task. The general procedure is to create the initial object and creating clones of it via operations not modifying vertex number and vertex structure. Because of this process it is very difficult to create basically different morph targets that keep these necessary conditions. The 3D active triangulation scanner developed at our Department can handle this problem, it can create morph targets about totally different real objects. Parametrisation of output targets can be wide, so the user has a freedom for modelling. Paper will draft basics of morph target creation, the structure of the scanner system and capabilities of the device and software.

Keywords: 3D scanner, morph, vertex, modelling

1. Fundamentals of Morphosis

The morphosis (morph) is a simple but attractive operation of 3D modelling. If two or more geometries (morph targets) has to be modified into each other, the morph operation can be used. The morph has two important necessary condition [8]:

- The vertex number of morph targets has to be the same
- The vertex order of targets has to be controlled (it is important that the modeller is able to control the vertex order)

These conditions are essential because of the functionality of the morph operation. It defines a connection between point pairs of two morph targets:

- vertex 1 of object A to vertex 1 of object B
- vertex 2 of object A to vertex 2 of object B, etc.

Vertices of resulting object (morph object) will be created as follows. These vertices will run on the edge connecting a vertex pair, the position can be controlled by a scalar, the morph phase. It can be a percentage value. If it is 0%, all vertices are at the starting position, the resulting object is the object A, if the value is 100%, the morph object is the object B.

If items of vertex pairs are not visually neighboring vertices but totally mixed in two morph targets, the operation can be made, but the process when morph phase is running from 0% to 100% can present uncontrolled geometry.

2. Configuration of Device

2.1. Components

The experimental device is an active triangulation scanner based on a vertically structured light source [1]. This device is built up as an experimental device to study the nature and the effect of the noise in the frame buffer and to elaborate the line following algorithm.

Main components of the scanner is the rotating table, the light source and the imaging device (camera) [2]. The layout is presented in the *figure 1*.

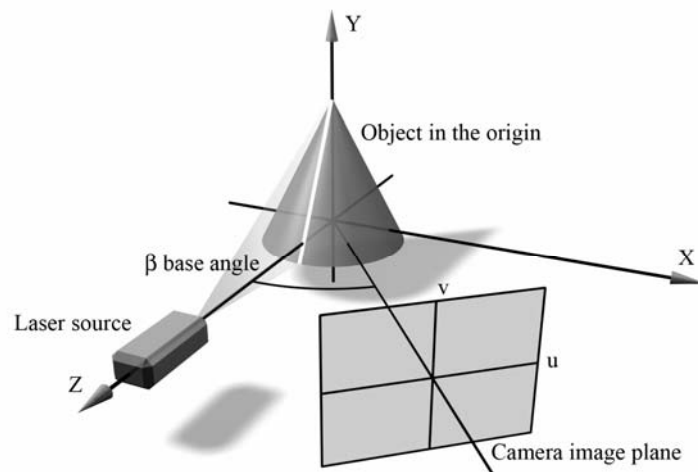


Figure 1. Layout of the experimental scanner system.

The object table can be rotating table to scan cylindrical objects or it can be sliding table to scan bas relief-like objects. The actual cross section parameter (rotating angle of shift of the sliding table) is set by step motor.

The light source is a 5 mW He-Ne laser. A cylindrical lens is used for shaping a vertical line projected to the object on the table. It is important that the intensity profile of the laser beam is Gaussian [3].

Formerly a 512x512 pixel resolution Visionetics frame buffer and a JVC TK-870E camera head were used. Now development is focused on to the progressive scan programmable digital cameras. First the most important topic is to choose the technology: which is the lowest noise camera, which technology is the best (CCD or CMOS).

2.2. Setting of the light source

This section presents how the light source has to be located in order to eliminate invalid line segments caused by configuration. Three kind of line configuration can be defined shown in the *figure 2*. Fully overlapping, partially overlapping and non-overlapping line segments can evolve. If a line segment is covered by other in full length, these are characterized as fully overlapping segments. If there are parts of both of segments that are common in a scan line, but there are uncommon parts as well, these are partially overlapping segments. If there is no scan line that is crossed by the two segments, these segments are non-overlapping.

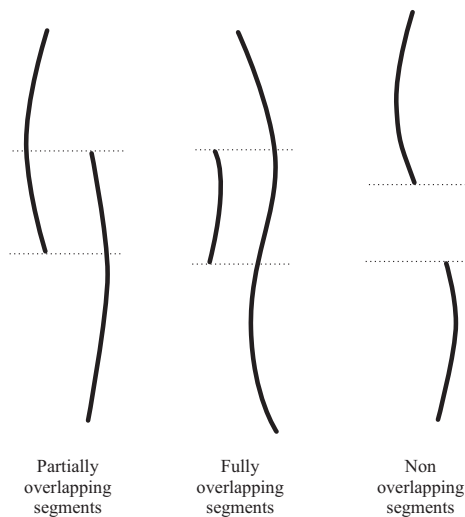


Figure 2. Relations of line segments in the frame buffer.

What is the good way to build the configuration up so that the frame buffer image contains at best one line cross section per scan line? In other words, based on the *figure 3*, how can be achieved $v_b > v_a$ relation?

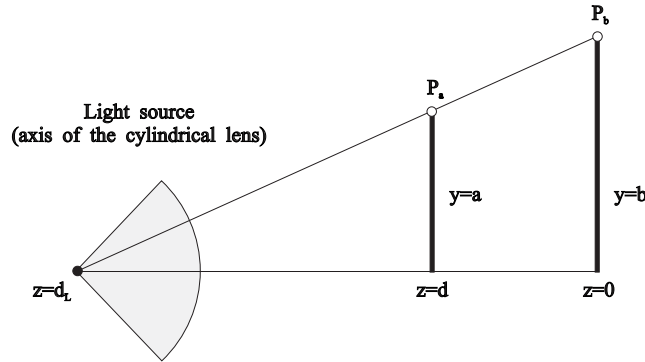


Figure 3. Description of geometrical parameters of light source location.

The target is to eliminate fully or partially overlapping segments. It can be proven that

$$\frac{d_c + f}{\cos \beta} > d_L \quad (1)$$

where d_c is the camera distance from the origin, f is the focus of the pin hole camera model, β is the base angle, d_L is the distance of the light source (the cylindrical lens) from the origin [12].

3. Model Generation

3.1. Structure of the System

The system communicates with the hardware via the **Driver**. Hardware components that has to be controlled are the imager system (the camera) and the rotating/translating table. The table parameter (angle or offset of the actual section) has to be controlled and the table has to be reset to the initial point. The driver offer some services for the calibration the system to the actual circumstances, like the setting of the sensitivity of the imager system, calibration of geometrical parameters of the system (camera distance, base angle, resolution,

etc.), setting of noise thresholds. The main structure of the system is presented in the *figure 4*.

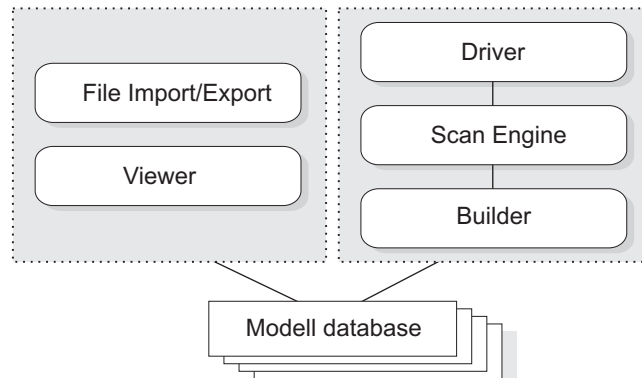


Figure 4. Structure of the system.

The **Scan Engine** controls the low level part of the process: the line following and segment clearing of the found line [9]. The system uses a unique subpixel line tracker, that can handle the noise of the frame buffer, and can control the accuracy of the line following in noisy environment [10][11]. After the line tracking it is necessary to maintain broken line segments and removing of invalid segments arisen by reflections. Process flow of the Scan Engine is presented on the *figure 5*.

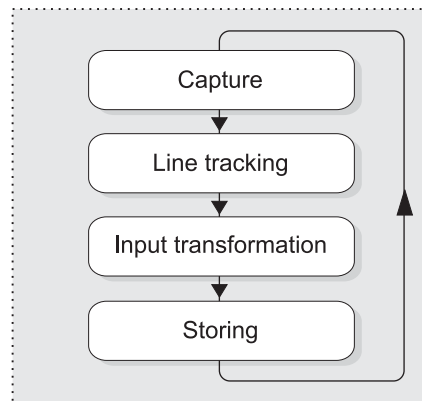


Figure 5. Flow of Scan Engine

The task of the **Builder** module is to build the model up. The continuous line has to be quantized. Next subsection will describe the available quantisation methods.

Separate lines (sections) are inserted into the model. Insertion has two steps:

- Insertion of vertices
- Creation of edges and faces

The model can be built up in some way. The input transformation is fitted to the input device configuration: different in case of rotating or translating tables. This information is stored in the database, however it is not necessary to build the model in the same way. If a bas-relief is scanned with a translating table, the model can be generated with a translating method (so the resulting model will be similar to the original object), or with a lathe operation, when the resulting object will be a cylindrical model. Or conversely, if an object is grabbed from the rotating table, it can be assembled with a lathe method (result: the model of the original object), or with applying the sliding method (section angles are transformed to offsets) a planar object can be rendered.

There some other additional functions of the builder. It can cap the top or bottom of the object (if it is not capped, it will be a tube-like model, if rotating table is used), or it can close or keep open a rotational model scanned with smaller then 360 degrees.

Models build with Builder up can reassembled in any time based on the drafted services, parameters. These models are in the **Model database**. The scanner software is a multi-document application, so we can work with more models.

The **Viewer** subsystem can show models and it provides a user interface for editing models, even the calibration and settings.

User can reach whole model or can edit cross sections of it, even order or geometrical parameters of them (angular or linear offset).

Some kind of visualisation method are under implementation. Model can be drawn with wireframe method or it can be shaded. Hidden surfaces can be switched on or off.

The **File import/export** module is responsible for the exporting models in well-known CAD formats, like 3D Studio ASC, 3DS formats, Autocad DXF format or native format of the scanner for later editing with full functionality.

Native format export/import is necessary for editability of scanned objects not only on the worksation where the scanner hardware is installed, but on any stand-alone workplace, where the same software is installed and configured as a stand-alone instance.

3.2. The Builder Module

The Builder gets raw sections from the Scan Engine. First it is quantized. After it the Vertex list is generated and the face list is calculated connecting current edge into the model. Present version of system can create only polygon mesh models, but no theoretical limits of creating NURBS surfaces further as well [4]. The draft of the Builder is shown on *figure 6*.

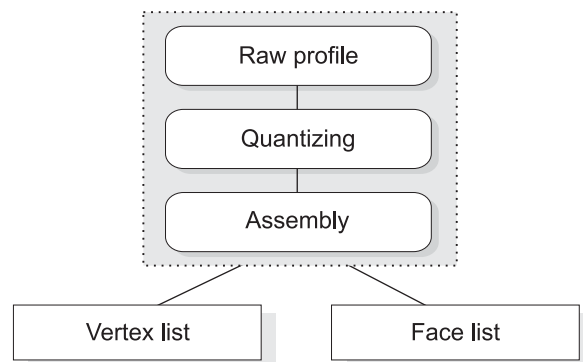


Figure 6. Process of the Builder.

Five quantisation methods are defined in the system. *Figure 7* shows these options.

1. The first called “*Native*” uses every pixels that Scan Engine produces. Vertex number is the function of physical resolution of the imager device.
2. *Equidistant distribution*. Vertex number of a section is depending on the length of the segment. Vertices will be distributed on equal distances.
3. *Fixed vertex number by equidistant distribution*. Start and endvertex of the line is fixed, remaining $n-2$ vertices is positioned at the same distant on the line. Distance between vertices is depending on the length of the line.
4. *Fixed maximal bias between strings and the detected line*. If a polyline is created based on vertices, maximal bias is defined between polyline segments and detected line. Vertices will be more dense if radius of curvature of the line is smaller, on nearly straight segments few vertices will be placed. In consideration of variable curvature vertex number and distances will be adaptive, depend on the local curvature.

5. *Fixed number of vertices by adaptive distances.* Start and endvertex of line is fixed, remaining $n-2$ vertices is distributed adaptively in the function of curvature of the line.

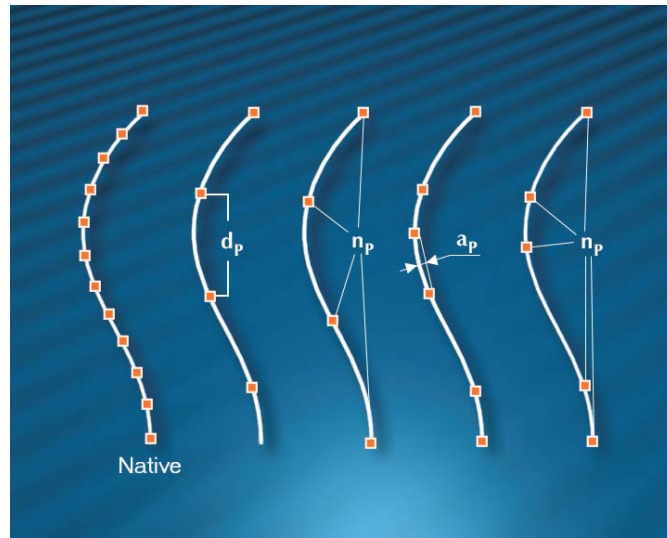


Figure 7. Quantisation methods of a profile line.

3.3. Scanning Morphable Models

The essence of the morphable model creation is in the quantizing of the cross sections. Two of quantisation methods described above are capable of creating morph targets, based on conditions described in *section 1*.

The vertex number similarity among morph targets can be provided at two points:

- Fixed vertex number has to be used with equidistant or adaptive distribution during quantisation,
- Fixed number of sections has to be captured.

In this way vertex numbers will be equal, and the vertex order will be the same because of the nature of described scanning method.

Figure 8 presents a morph target pair of computer mouses.

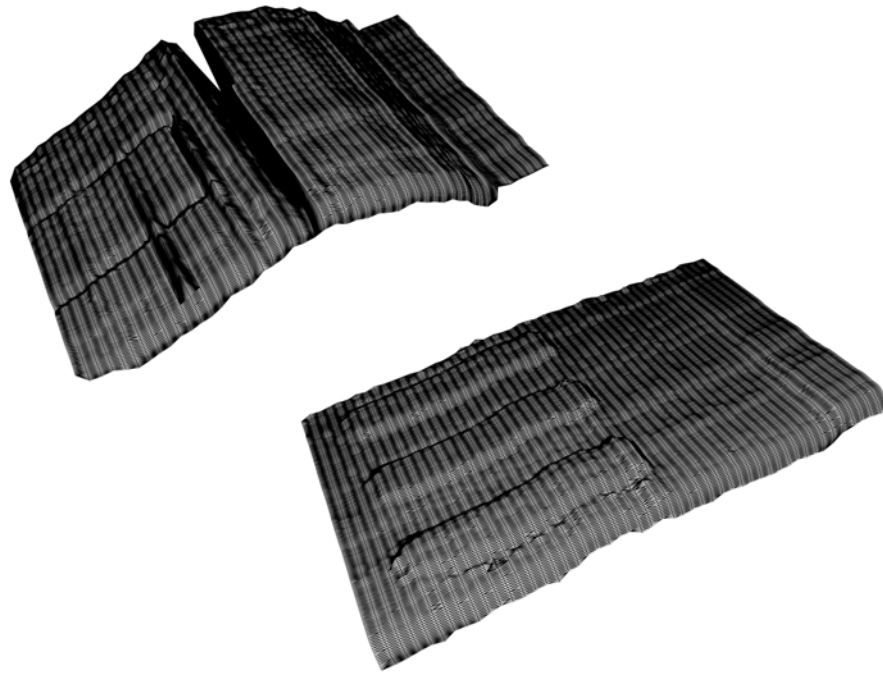


Figure 8. Two computer mouse surface as morph target objects. The first has a scanning error caused by some reflection, but the system can create morphable targets in this case as well.

Conclusions

The active triangulation scanner described before offer an easy way to create morphable object. In this way during reengineering tasks similar workpieces can be compared, differences can be analysed [5][7]. In human movement analysis (sport or medical applications) bone system-based characters can be scripted and skinning can be made by morphable surfaces targets. For instance it can be a good application in the future if some slowly deforming organs are scanned periodically, and the changing of geometry can be studied in movement in 3D at a later phase. This approach can be applied in industry at degrading machinery components comparing a new component with the used one by 3D morphosis [13].

System development is under processing, now applications are searching for. The main target was the accuracy of the geometry, so a lot of application can be found in the industry and bussiness life [6].

References

- [1]. Yoshiaki Shirai: Three-Dimensional Computer Vision, Springer Verlag, 1987.
- [2]. Jorge L.C. Sanz: Advances in Machine Vision, Springer Verlag, 1989.
- [3]. John E. Harry: Industrial Lasers and Applications (Ipari lézerek és alkalmazásuk, in Hungarian), Műszaki Könyvkiadó, 1979.
- [4]. I. Vakalis (Italy): Digital Topology: A survey of 3D object representation and its configuration space, 3th WSEAS CSCC (WSEAS International Multiconference on Circuits, Systems, Communications and Computers), Athens, Greece, 1999.
- [5]. Marc Rioux: Applications of Digital 3D Imaging, Canadian Conference on Electrical and Computer Engineering, 1990.09.03-06., pp. 37.3.1-37.3.10
- [6]. Terry T. Wohlers: 3D Digitizers, Computer Graphics World, 1992.07., pp. 73-77
- [7]. Francois Blais, Marc Rioux, Jacques Domey: Optical Range Image Acquisition for the Navigation of a Mobile Robot, IEEE International Conference on Robotics and Automation, 1991.04.09-11, pp. 2574-2580
- [8]. Alan Watt, Mark Watt: Advanced Animation and Rendering Techniques, Theory and Practice, Addison-Wesley, 1992.
- [9]. Kae-Jy Chou, Brian G. Schunck: Nonparametric Algorithm for Edge Localization, IEEE Computer Vision and Pattern Recognition, 1993., pp. 795-796
- [10]. Tibor Kovács: Accuracy Analysis of an Edge Following Algorithm in Noisy Frame Buffer, IEEE ACCV'95 Second Asian Conference on Computer Vision, Singapore, December 5-8, 1995, Proceedings Vol 2, pp. 430-434
- [11]. Tibor Kovács: Accuracy Prediction in a 3D Active Triangulation Scanner, Machine Graphics & Vision (International Journal of the Institute of Computer Science, Polish Academy of Sciences) vol. 10, no. 1, 2001 pp. 75-87
- [12]. Tibor Kovács: Efficient Line Following Algorithm with Controlled Noise Dependency, WSEAS Conference on Automation and Information (ICAI '04), 2004, Venice
- [13]. Crispin Littlehales, Marc Rioux: White Light Magic, IRIS universe, the magazine of visual computing, number eighteen, Siliicon Graphics Inc., 1992., pp. 24-27