# Content Acquisition Techniques for 3D Visual Computing

Dinesh Shikhare
National Centre for Software Technology,
Juhu, Mumbai 400049, India.

**Abstract**

Content creation for 3D graphics systems, virtual reality walkthroughs and interactions and multimedia presentations has been known to be a labourious and difficult problem. A lot of research is being carried out towards rapid generating detailed and realistic 3D geometric data, 2D images for textures to be mapped on them. This report present a brief survey of state of the art of various data, model and interaction aquisition techniques currently in use for 3D Visual Computing. More stress is put on representation and processing of the data rather than the physical processes and gadgets used to obtain the data.

# 1 Motivation

Last few years have seen a rapid improvememnt in the performance of graphics hardware. It is now possible to obtain PCs having fairly good 3D graphics capabilities and develop interactive graphics applications with reasonably complex geometric models and sophisticated rendering algorithms. This development is largely due to: (a) improvement in the processing speeds of the processors in these machines, (b) enhanced architectures for systems which allow high speed 3D graphics, e.g. Accelerated Graphics Port (AGP) architecture on PCs, multi-pipeline rendering hardware on SGI machines and (c) implementations of powerful 3D graphics APIs and their efficient implementations which take advantage of the underlying hardware.

Creating compelling and detailed graphic content to take advantage of powerful hardware and software platform has been the focus of research in graphics community in recent years. Traditional techniques for generating content have been extremely expensive and labourious. Most of the software tools created for generation of 3D models and textures expect humans to enter the details of the 3D models manually using the graphical user interface of the packages. Current efforts have concentrated on automatic capturing of detailed geometric objects by direct observation of the real life objects. These techniques "scan" 3D objects and their attributes to obtain sufficient information to render them in a digital environment.

In this report, we survey and compare these techniques in the following aspects:

- Captured information
- Preprocessing and filtering required
- Storage requirements
- Survey of available literature

# 2 Survey of model acquisition techniques

## 2.1 Modeling packages for polyhedral and smooth surfaces

Modeling packages like 3D Studio, Alias/Wavefront, etc. have been used extensively to create geometric content. These packages provide a comprehensive set of tools to create complex geometric shapes using geometric primitives. Various modeling techniques are supported commonly:

1. **Constructive solid geometry (CSG):** This technique is used to build complex geometric shapes from primitives like rectangular parallelopiped, sphere, cylinder and their transformed shapes. The tools to combine these basic shapes to obtain complex shapes are boolean operations like intersection, union, subtraction of shapes.

2. **Modeling of smoothi surfaces:** To model smooth surfaces, these packages provide facilities to create and edit polynomial surfaces such as Bezier patches, B-Spline or NURBS surfaces. These surface are defined using control points describing the control mesh of the surface. The shape of the surface is determined by the positions of the control points.

3. **Specialised techniques:** Some special techniques like fractal-based ot graftal-based modeling are used to model structures found in nature such as mountains and terrains.

4. **Attribute Editing:** To edit attributes like material properties and texture, these packages provide specialised tools. The tools are often simple 2D drawing and image processing facilities.

5. **Miscellaneous tools:** Various other techniques such as modeling sweep surfaces by sweeping a shape along a path, extrusion of a 2D shape, application of modeling transformations on simple 3D geometric shapes, etc. are very common.

While using these facilities, the users must have fairly good artistic capabilities. The use of these packages to create substantially complex and realistic models is non-trivial and laborious.

Since these models are "hand-crafted," very little or no preprocessing needs to be done before using them in 3D applications. The only processing that may be needed is obtaining the model at various resolutions to ensure that the applications render these models in reasonable time.

## 2.2 3D digitizers

To obain 3D digital models of existing real world objects or to obtain such data from clay models, 3D stylus based digitizers are commonly used. Figure 1 illustrates one such digitizer. The mode of usage for such digitizer involves the following steps:

1. Create a rectangular grid of points on the 3D surface to be digitized

2. Place the object in the range of stylus and initialize the digitizer so that the frame of reference is fixed

3. Digitize (record the 3D positions of) the points on a patch of the surface in a systematic row or column order. Digitizing each point involves placing the stylus on the point in the grid and clicking the "record" button.

Figure 1: A 3D digitizer

Once this data is obtained, various software techniques are used to either retain the digitized data as a collection of bilinear patches or a higher order surface is fit on to this data. As a result, one obtains a piecewise continuous patches of the surface describing the model being digitized. Besides this pre-processing stage, often there is a need to filter and smooth the data to correct the human error

## 2.3 3D Scanners

Laser range scanning devices based on light interferometry provide a much more automatic tool for obtaining a digital model of an existing 3D object. These scanners are rotated around a given object at a specific step size and a large number of measurements are taken. These measurements often consist of geometric position as well as the texture information at the scanned point. The scanning process gives a complete description needed to render the object. Figure 2 shows an example of a whole body scanner that can produce a detailed description of human sized objects.

Like the previously discussed 3D digitizer, this technology also returns a set of systematically row-wise or column-wise scanned points. Depending on the application, the data is converted to either a triangle mesh or a quadrilaterals providing a piecewise bilinear model of the surface.

The data automatically scanned by range scanners tends to be extremely verbose. Even if the scanned surface is mostly flat in many places, a large number of points are recorded to describe the part. This unrequired combinatoric complexity of the model needs to be reduced by polygonal mesh simplification methods. A large number of people have been researching the area of mesh simplification.

Figure 2: Whole body range scanner

## 2.4 3D Range Scanners

Range scanning devices are basically digital cameras that capture, for each pixel, the distance of the closest object. A single snapshot, however, gives the 3D information only from one viewpoint. To capture the 3D geometry from all directions around the 3D object of interest, multiple pictures need to be take with substantial overlap across the images. Such range images are combined to form a new complete mesh representing the scanned object.

The merging process is non-trivial. Greg Turk and Mark Levoy [7] present their algorithm for reconstructing the mesh. The basic algorithm is described for merging two scanned patches obtained from two range images having a substantial overlap. Their algorithm consists of two distinct phases:

1. Obtaining a triangle mesh from a range image
2. Merging two triangle meshes which have an overlap. The overpping parts of the meshes are trimmed against one another along the boundary of overlap. The unwanted pieces are thrown away.

The algorithm uses many geometric algorithms over the domain of 3D triangles. These algorithms are mostly used to carry out robust intersection of triangles and triangle meshes to obtain the curve of intersection and a new local triangulation.

## 2.5 Image based acquisition of 3D models and textures

Recovery of 3D geometry from a set of images snapped from known or unknown viewpoints has been a problem of interest in the Computer Vision field for a long time. In recent years computer graphics community has explored the area to acquire 3D geometry data from images.

The three subproblems commonly solved by all the approaches can be abstracted out as:

1. Obtain a point correspondence (or more generally, feature correspondence) across images of the 3D object being reconstructed. This specification of correspondence also defines the connectivity of the points.

2. Solve camera-pose estimation problem to obtain the 3D relative coordinates of camera and subsequently the matched points.

3. Map the appropriate textures for getting correct view-depepndent rendering of the 3D objects.

Paul Debevec et al [1] present a hybrid geometry and image based approach to model architectural monuments. To capture geometry from multiple photographs, these techniques use photogrammetry methods. Formulate a complete photogrammetric problem, one needs to identify a point correspondence for at least 5 points across two images. The model is obtained by solving the photogrammetry problem. The associated textures are captured from images. Viewing these models from arbitrary viewpoints is now possible since the 3D model is available. However, the texture mapping must be view dependent. This is because, the textures captured from images of the 3D objects will not have enough information about the occluded geometry. Hence as the viewpoint changes with respect to the reconstructed geometry, the model's attributes like texture need to be dynamically changed.

Some other authors have reported model-based methods for reconstruction of 3D objects. Rockwood and Winget [5] report such method which starts with a connonical model of the target shape to be reconstructed. For example, to reconstruct a vase from images, the connonical model would be a cylinder mesh. An error function is formulated between the rendered image of the current approximation of the model and the photographs. Simulated annealing algorithm is used to purturb the points on the mesh of the model until the rendered version of the model is close to the photographs of the 3D object being reconstructed.

## 2.6 Image-based rendering

In image-based rendering techniques, no attempt is made to reconstruct the 3D geometry of the actual 3D world objects, but the captured images are warped, transformed, clipped and pasted in appropriate places to generate an imagery that gives an illusion of 3D scene navigation as the viewing parameters of the viewer change. The aim is gather enough data about the model to be able to create a reasonably realistic reproduction of the 3D object from an arbitrary viewpoint and possibly different illumination conditions (see [8]).

Image mosaicing approach of [6] constructs a panoramic view of 360 degrees around a viewer for a fixed position to give a limited virtual reality experience. The viewer is made to believe that he/she is viewing a 3D world around a fixed position, iteractively. However, actually what is rendered is a mosaic of images mapped on a cylinder or a sphere. Construction of a panoramic single image from a set of overlapping images of a scene is the contribution of Szeliski's work.

Lumigraph demostrated by Gortler et al [2] is another example of image-based modeling and rendering of 3D objects. This approach captures a set of images of a 3D object from many positions around it. Based on the position of the viewer, an interpolated version of a set of images related to the viewpoint is displayed to the viewer. The paper presents the formulation of continuous version of Lumigraph, but discusses an implementation of discrete model for achieving real-time rendering of Lumigraph.

Plenoptic modeling approach presented by Leonard McMillan and Gary Bishop [4] presents an algorithm for view depenedent rendering of models captured as range images and texture images. This work, however, requires additional data of range images to handle depth information, while the positive aspect of this work is that fewer images are required to reconstruct the views of the captured model from other viewpoints.

## 2.7 Volumetric data visualization

3D data is also obtained as discrete volumetric data by MRI scanning or computed tomography (CT-scan). Visualization of this data is a common requirement in medical imaging (see Figure 3) and industrial non-destructive testing applications.

The data is represented as a 3-dimensional array of scalar or vector values sampled at discrete points in a 3D volume. Very often the captured data requires preprocessing of the following kinds:

1. Filtering of noise and irregularities: This data is often filtered by using some spatial filter like the Gaussian filter to remove the measurement errors and spikes.

2. Subsampling of dense data: The captured volumetric data is too dense to load in the memory at one time. To be able to interactively visualize the volumetric data, a low resolution of version is obtained by subsampling.

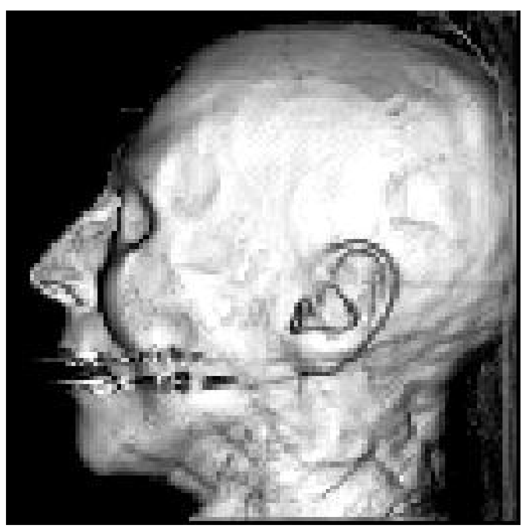

Figure 3: A 3D digitizer

Various approaches exist for visualization of the volumetric data. The requirement of visualization, however, is uniform across these approaches: the volumetric data needs to be visualized as an iso-surface at a particular threshold value of the selected scalar attribute.

The two major approches for visualization are:

1. Construction of a triangle mesh of the iso-surface using algorithms like marching cubes [3] or marching tetrahedra.
2. Visualization by shooting rays through the pixel to be rendered into the volume data and accumulating the intensities between the threshold value's crossover points.

# 3 Conclusions

In this report we have presented a survey of representative techniques of obtaining 3D content for visual rendering. Various representational and algorithmic issues in capturing and using the data have been highlighted. In our future study, we intend to concentrate on image-based modeling and rendering of 3D environments.

# References


[1] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH ' 96)*, pages 11–20, 1996.

[2] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '96)*, pages 43–54, 1996.

[3] W. E. Lorenson and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '97)*, pages 163–169, July 1987.

[4] Leonard McMillan and Gary Bishop. Plenoptic modeling: An image-based rendering system. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '95)*, pages 39–46, 1995.

[5] Alyn Rockwood and Jim Winget. Three-dimensional object reconstruction from two dimensional images. *Computer-Aided Design*, 29(4):279–285, 1997.

[6] Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '97)*, pages 251–258, 1997.

[7] Greg Turk and Mark Levoy. Zippered polygon meshes from range images. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '94)*, 1994.

[8] Tien-Tsin Wong, Pheng-Ann Heng, Siu-Hang Or, and Wai-Yin Ng. Image-based rendering with controllable illumination. In J. Dorsey and Ph. Slusallek, editors, *Rendering Techniques '97 (Proc. Eighth Eurographics Workshop on Rendering)*, pages 13–22, New York, NY, June 1997. Springer Wien. Available from: http://www.cs.cuhk.edu.hk/ ttwong/papers/lightfld/lightfld.html.