Invited key-note paper at the International Conference on Shape Modeling and Applications, Aizu-Wakamatsu, Japan, IEEE Computer Society Press, pp. 64-70, March 3-6, 1997.

The 3D Revolution: CAD Access for All!

Jarek Rossignac
GVU Center & College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280, USA

Abstract

The manufacturing industry has invested vast amounts of resources in the deployment and use of solid modeling technology. Although expensive to generate and potentially very valuable in many product related activities, 3D models have rarely been exploited to support product management, documentation, collaborative review, and promotion, because they were only accessible to trained designers equipped with expensive graphics workstations. Intranet access, popular 3D exchange formats, and affordable 3D graphics chips permit to download and view 3D models using a personal computer. Although these basic capabilities are revolutionizing the entertainment and marketing industry and have reduced the cost of a design station, they are of little help to non-designers in the manufacturing industry. The author articulates a vision where 3D data is available and exploited at all phases of a product life cycle. The paper investigates the shortcomings of the current technology, identifies the fundamental research issues, and reviews recent advances in 3D data compression, in the automatic generation of levels-of-detail for interactive rendering, and in the innovative exploitation of 3D input devices for an intuitive and effective navigation.

.

1. Who needs to access 3D data?

Today, CAD models are successfully exploited for the early detection and correction of design errors, for the automation of manufacturing and analysis tasks, for consistency control, and for the generation of blueprints and their electronic distribution to suppliers. The deployment of CAD systems—and in particular of three-dimensional solid modeling technologies—within the manufacturing industry represents an investment of hundreds of billions of dollars in equipment and software and significantly more in support and labor costs. Over the years, the principal asset in CAD technology for a manufacturing company has shifted from hardware to software, to design skills, and finally to the data itself. For instance, the creation of a CAD model for the assembly of a complex engine may require over a hundred person-years and will be used to control and automate many downstream applications during the product's life cycle. The 3D data has more value than the equipment and software used to create it.

Because the exploitation of CAD data requires highly specialized design skills, the result of the enormously expensive design phase is rarely used for marketing purposes or for the creation of illustrations for manuals and training material. Very often it is more effective for a documentation specialist to recreate an approximating exploded view of a subassembly using 2D illustration tools than to have a designer generate exact views from the master 3D model and to import the result into the document preparation software. More importantly, product data management (PDM) systems have very limited access to the 3D model. Requests for engineering changes must use words—or at best red line mark-up over electronic copies of blueprints—to indicate precisely which feature or detail in the product should be affected by the change. Access to the 3D model and support for 3D annotation and mark-up would significantly increase the communication bandwidth and reduce the probability of misinterpreting ambiguous descriptions. Unfortunately, the majority of people involved in requests for engineering changes and the associated approval and verification steps are not designers and do not have access to a CAD system. Furthermore, the integration of CAD and applications suffer from complex software legacy and intellectual property problems.

We believe that the design cycle for complex assemblies could be significantly shortened and the resulting product and manufacturing plan be considerably improved by the integration and deployment of 3D viewing and annotation technologies in all product-related communication and documentation tasks. An engineering change or problem report could simply say "**This** slot is too wide and **that** blend should go all the way to **here**", where **This**, **that**, and **here** refer to 3D selections or locations that can be easily created and viewed by professionals who have no expertise with CAD systems and no easy access to one.

2. CAD access for all

In this section, we outline a utopic vision for a product design cycle fully supported by a shared 3D database. In the next section, we discuss the difficulties in implementing such a vision and we identify key research challenges.

2.1. Network access

The 3D databases capturing the latest updates to the product should be available for privilege-controlled access throughout the enterprise, and whenever necessary, to its customers and suppliers. The database should not only capture the latest version of each product or family, but should also identify all known and unresolved design problems and should capture the design history and rationale. The database should also identify the teams responsible for the various components and support queries and filters that reflect the different ways to organize and view the product. For example, an individual may wish to view all the electric wires and pipes of a specific section of an airplane or factory and retrieve all engineering changes that addressed safety problems.

2.2. 3D support for communication

Most design and manufacturing issues evolve around specific features in a single part or in an assembly. Problem identification and conflict resolution activities cannot be carried out without a graphic representation of the product. Working with 2D drawings is often less effective than interacting with the real parts or with 3D models, especially for people not accustomed to drafting practices. In the ideal situation, all product related communication should be supported by 3D models. For example, users may attach information to specific features of a model or refer to them in the text of a report just by pointing to them on a graphics screen. Users who have downloaded the model or the report should be able to read the annotations associated with particular geometric features or to read the report and immediately see what the authors were referring to.

2.3. Shared 3D environments

Live discussions between coworkers at remote locations should be supported via shared 3D environments, so that a distributed team working on a virtual model could be just as effective as a co-located team working around a physical mock-up model. Team members should be able to see who is pointing to what and share their views of the model.

2.4. Integrating 3D with other applications

3D viewing and referencing capabilities should be integrated with other personal productivity tools. As images can today be dragged and imbedded in other text processing and page composition applications, 3D models should be available for integration with other media. For example, a documentation specialist should be able to get the desired subassembly models from the shared database, imbed them as an illustration in an electronic document, set up one or several views to be referenced by links from the document, and save it. Ideally, revisions to the relevant parts should be reflected automatically whenever the electronic documentation is viewed.

3. Why is it so difficult?

The scenarios discussed in the previous section are far from revolutionary and at first glance appear easy to implement using existing technologies. Furthermore prototype or even product implementations of some of these concepts have been already demonstrated. Yet, none of these possibilities have been effectively deployed in a production environment with a clearly positive impact on productivity.

Implementing a simple scenario where employees with no expertise in CAD use existing 3D models to support their activities and communication needs requires solving several challenging problems: performance issues stemming from the complexity of industrial datasets; usability problems linked to the inherent difficulty of 3D navigation and to the diversity and complexity of CAD user interfaces; and the difficulties of integrating 3D access software with personal productivity tools and within the overall design process of a corporation.

3.1. Complex models take forever to download

Today, the model for a complex assembly (such as an engine, a car, or a plane) may have over 100,000 parts. Although the parts' geometry may be represented with considerable accuracy using curved surfaces, it is often tessellated for rendering purposes. Hardware-assisted rasterization is particularly effective at rendering triangles and because tessellating curved surfaces is computationally intensive, typically tessellated models are generated once and saved for graphics. A polyhedral approximation of a single part that is sufficient for most viewing purposes would on average contain 500 vertices and 1000 triangles. Of course, these statistics depend considerably on the application domain and on the granularity of the model. Nevertheless, they capture the order of magnitude in the complexity of industrial models and help understand why modeling and data transfer solutions that were developed for simple scenes used in entertainment applications are not viable in industrial settings.

Popular format for representing triangular meshes with the associated photometric data (normal, colors) require close to a hundred bytes per triangle. At this rate, a graphics model of a complex assembly may require around 10 Gigabytes of storage. During intensive design, the assembly model is updated periodically. Advanced caching techniques may be devised to avoid dispatching subsets of the model that have not been modified and to load onto a user's computer only those portions of the assembly which are needed for rendering. Still, interactive performance will require both a large amount of local storage and a high bandwidth communication channel. For example, suppose that only one percent of the data is visible at each frame and that only 0.1% is changing at each frame. We would need 100 Megabytes of local storage and a download of 10 Megabytes several times per second.

The solution of course is to use 3D compression. The compression scheme should be sufficiently rich to handle all the necessary information, should be suitable for efficient coding and very fast decoding, and should be lossless, or at least should provide good estimates of the uncertainty resulting from lossy compression.

3.2. File formats are not compatible

Although the triangular meshes used for graphics are very simple structures, a surprisingly large number of representation schemes have been developed for storing and processing them. Some of these representations capture topological and differential continuity aspects of the original model that cannot be extracted from a simple list of triangles. Are two adjacent triangles part of the same smooth surface or is their common edge a real sharp edge in the original model? When two parts of a polyhedral mesh are touching at a vertex, are they simply in contact or did the model capture a non-manifold topological situation that is intrinsic to the structure of the mesh? Although standards for representing triangle meshes and more complex geometries have emerged, the various CAD systems which are used in the manufacturing industry and that can generate tessellated models differ significantly in their interpretation of the semantics and validity rules for polyhedral meshes. Consequently, topological and continuity information may often be lost, which in turns make model verification more difficult. Consider for example a cylindrical hole with a cylindrical pin that barely fits the hole.
The original curved surface model may be used to verify that the pin has two degrees of freedom. However, should the pin and the hole be approximated by polyhedral models, the tessellation will typically prevent the pin from rotating and may even result in assembly interferences. Increasing the precision of the faceted model will rapidly increase the number of facets and will often only defer the problem.

The solution seems to call for a clever coding scheme that captures in a minimum number of bits the geometry, the topology, and the continuity or other surface properties.

3.3. Casual users cannot navigate in 3D

The view parameters (i.e. the way a 3D model is projected onto the screen) may be controlled one by one or in groups. Various mappings have been devised to interpret the users' gestures or interactions with the various input devices. Although some solutions strive to support an intuitive metaphor (for example spinning a virtual sphere centered around the object), there is no accepted standard that would permit for a casual user to migrate from one 3D system to another. Virtual reality, which targets the ultimate match between our natural view controlling skills and the response of a 3D viewing system, often requires delicate and expensive set-ups, not always suited for continuous usage in industrial settings. Furthermore, the natural interaction paradigm may lead to considerable loss in productivity. Imagine for example reviewing the model of a large factory and having to walk from one end to the other. Whatever input device is used to detect the walking motion, the "walk where you want to go" approach seems better suited for indoor exercise than for design inspection. Switching between "model in the hand" and "immersive" metaphors may leave the user disoriented.

The solution should provide a view control mechanism so trivial to understand that a first time user should be comfortable with it in a few second. More importantly, the mechanism should quickly make the user productive and permit to fully concentrate on the design and not on the user interface. The solution should also be suitable for inexpensive desk-top systems and for collaborative environments. Furthermore, the same paradigm should be used for controlling the relative position of the geometric features in an assembly as for controlling the view.

3.4. Rendering a complex model takes too long

An effective interactive manipulation of the view or of the geometry is heavily dependent on realtime graphics feedback. The slightest delays in the systems response confuse the user and make humans less productive. They also distract the users attention. Ideally, the graphic feedback should not be delayed by more than 7/100 of a second. Until recently, the relation between the cost of a workstation and the number of triangles it could display at these interactive rates was $1 per triangle. A $20K workstation would display about 300,000 shaded triangles per second and hence support interaction with models of 20,000 triangles at 14 frames per second. Such models would typically contain about 40 simple parts, which is sufficient for designing a single component in the context of its surroundings, but is not appropriate for inspecting or annotating a larger assembly. Although the ratio between the hardware cost and graphics performance is dropping rapidly, we still have a very long way to go before the user of a $2000 desk-top system is able to comfortably interact with 100 million triangle assemblies. The discrepancy will be further exacerbated as the detailing of the CAD models increases. Today, many detailed features, such as holes and fasteners are still not modeled, because of storage and performance constraints. This omission leads to serious difficulties in detecting design errors. Progress in creating more precise models will exacerbate the rendering problem.

The solution is to avoid doing unnecessary work during graphics and to trade visual precision for interactive performance whenever necessary. For example, internal components and backward facing triangles that are not visible from a given view should not be rendered. Furthermore, small details that cover a very small area of the screen may require rendering large numbers of triangles with very little perceptible effect on the overall picture. It may be desired to skip some of the details, at least during interactive manipulation of the camera or of the objects. We need effective techniques that can quickly produce approximate images of highly complex 3D scenes. A particularly popular approach to this problem is the precomputation of several levels of detail for each component. Each level contains less triangles than the previous one and may be used as a cruder visual approximation of the original shape. Lower levels of detail are typically used for objects that appear small on the screen. The difficulty lies in the fact that the precomputation needs to be very fast and fully automatic (since it must be routinely applied to thousands of models as they are created or altered). Furthermore, the computation must be reliable and should not be affected by the variations or flaws in specific modeling schemes. For example, simplification algorithms developed to generate levels of details of manifold polyhedral meshes are of little use for CAD models that exhibit non-manifold situations or for architectural models which combine closed meshes with open surfaces (manifolds with boundary) and with lower dimensional elements (curves, surfaces).

3.5. Drag-and-drop does not work for 3D objects

Although a particular 3D viewer may be compliant with current standards for object linking and embedding or with the internet browsing tools, users should have a direct access to the individual objects in an assembly, and not be limited to the granularity of a file. One may open an entire assembly model, find the appropriate subset, create a shadow for it in a local environment, construct exploded views or assembly animations, and save the result together with the document in preparation. The actual use of each object in the local application may be affected by subsequent design changes. For example, if a local configuration is stored in terms of local coordinates for each part, the replacement of one part in the master model by another one that has a different local coordinate system will invalidate the local configuration.

3.6. Persistent references in evolving models

Annotating a face of an assembly or referencing a geometric feature from within a body of text result in the creation of a link between the text and a set of geometric entities of the 3D model. What happens to these links as the geometry is altered? Simple alterations that change a few dimensions but do not affect the topology of the representation may preserve the references. More complex changes that affect the number of vertices and faces may invalidate references to these faces. A special case of interest is the preservation of references during the construction of levels of detail. An even more problematic situation is the replacement of a part by another similar, but independently designed part.

Ultimately, we should strive to provide mechanisms for representing references that may be recomputed automatically from higher-level specifications and that would produce intuitively correct results. Because inferring the user's intent from a simple gesture is practically impossible and because we cannot expect the user to "program" a reference-evaluation function for each reference, we suggest to address the problem by first supporting a dependency graph which will be used to flag all references that may be affected by a particular change and thus ask the user to verify the reference, and then to try and infer new references automatically by learning from the history of user provided matches.

4. Recent progress

We review in this section several recent advances that contribute to the overall vision of "CAD access for all". The specific solutions discussed here have been developed at IBM by the author in collaboration with colleagues in the IBM T.J. Watson Research center. Furthermore, these research innovations have been expanded into product quality solutions and have been successfully integrated into IBM's design review product for highly complex assembly models: the IBM 3D Interaction Accelerator [1]. The details of these solutions have been published elsewhere. We focus here on their characteristics, on their impact on our vision, and on the opportunities for further research.

4.1. Compression

The geometric representation of a 3D shape suitable for rendering is typically a variation of a triangle mesh. Some-times, more complex polygons are used, but they are ultimately triangulated prior to rendering. Efficient triangulation algorithms exist [2]. It is therefore appropriate to consider only triangle meshes. A simple representation of a triangle mesh stores the vertex coordinates (maybe as an array of floating point triplets) and the definition of the triangle (maybe an array of integer triplets, each being an index onto the vertex array). Photometric information, such as the normal direction at vertices, colors, and texture coordinates for shading calculations are associated with individual vertices or with triangles, or more precisely with vertex uses for each triangle. This association may be stored in simple datastructures using redundant entries. Because popular shading hardware had limited internal memory for vertices, a given vertex may be sent and processed by the graphics subsystem several times (often twice or more and at most once per abutting triangle.)

Two directions for compression have been investigated. The first one attempts to compress the geometry (i.e. the vertex coordinates) by using lossy quantization, predictors, and entropy encoding. Reported geometric compression techniques with acceptable loss for most rendering conditions may reduce the storage for vertices and normals by a factor of 3 or 4. The second one focuses on compressing the definition of the triangles and on capturing the topology of the mesh. The 3 vertex indices associated with a triangle would occupy 12 bytes if stored as integers. The approach developed by Taubin and Rossignac [3] reduces this requirement to less than 2 bits per triangle. It is based on the construction of an extended version of a vertex spanning tree that cuts the triangle mesh into a flat triangulated polygon. The cut and the polygon are efficiently encoded as two dual trees. The triangulation of the interior of the polygon is encoded using one bit per triangle which indicates which of the next two vertices should be used to construct the next triangle. Fast compression and decompression algorithms have been developed. Because the spanning trees organize the vertices in terms of their geometric proximity, they provide good ancestors for predicting vertex positions and thus improve the results of geometric compression. Although Taubin and Rossignac's approach has been extended to non-manifold meshes with boundaries, several open issues remain. A few examples are provided here. Independent compression of the component in an assembly may alter the topology of their contacts because of the errors in vertex coordinates. Lossy compression with absolute error bounds may significantly alter small objects or features beyond recognition and have little compression effect on large objects. On the other hand, relative error bounds may alter relatively small features of large objects and affect the way they interact with other smaller objects. Compression schemes that convert from one representation to another may loose information or may break links between auxiliary references and the geometry they refer to.

4.2. Simplification

The author has explored two complementary simplification techniques, which automatically compute level-of-detail representations for triangular meshes. In both cases, polyhedral faces are triangulated prior to simplification.

The first approach developed by Borrel and Rossignac is based on vertex clustering [4]. Vertex coordinates are quantized (using integer truncation or possibly more advanced clustering techniques). Vertices with identical quantized coordinates form a cluster and are replaced by a unique vertex, selected to be the best representative for a cluster. (A good choice is the original vertex closest to the center of mass of the cluster vertices weighted by visibility and sharpness factors.) Then, the triangle mesh is simplified to remove degenerate triangles (i.e. triangles that have two or three identical vertices) unless they are not adjacent to other valid triangles. The advantages of this vertex clustering approach is its efficiency (simplification amounts to coordinate truncation, vertex averaging, triangle sorting, and coordinate comparison), its robustness (no assumption is made as to the topology of the original mesh), and its effectiveness (small disconnected features are often merged to produce larger simplified ones). The drawbacks of the approach lie in a suboptimal simplification. For a given error limit, the resulting triangle count may exceed what slower simplification techniques can yield. The approach also makes it difficult to control the exact triangle count in the simplified model. Nevertheless, this is the only simplification technique that has been reported to perform well in industrial settings (mostly because of its performance and reliability, and of course because of the simplicity of the underpinning algorithms).

The second simplification technique, which collapses edges one by one has been developed by Ronfard and Rossignac [5]. Candidate edges are sorted so that we first perform those collapse operations which least affect the pointset covered by the mesh. The incremental approach is well suited for controlling the triangle count or the maximum error for each level of detail. This approach is more complex to implement, slower than vertex clustering and more sensitive to topological inconsistencies in the original mesh.

Other simplification techniques have been surveyed by the author in [6].

Most important research challenges in this area address the convergence of simplification and compression technology in a form that is suitable for realtime extraction and rendering. Furthermore, the exploitation of a few static levels of detail may be problematic for relatively large and complex objects, especially when one end of such an object is closer to the viewer and needs to be seen in detail while features in other distant parts of the same object appear small and should be rendered using lower levels of detail. Several techniques that compute multi-resolution models have been recently proposed. Because of their overhead they should be compared to simpler techniques which split the large object into smaller components.

4.3. 3D navigation with a virtual camera

The virtual camera developed at IBM research is particularly well suited for a novice or casual user with no a priori expertise with CAD or with interactive graphics systems, because it enables anyone to become, within 30 seconds, entirely comfortable with the view control and capable of navigating through a virtual model. Furthermore, the virtual camera provides a significant productivity boost for most view manipulation tasks.

The virtual camera setup is based on a 3D model and a floor plan of a construction site, a building, an airplane or a ship. The current view of the model is displayed on a large rear projection screen (possibly in stereo) for group review or on a simple monitor for individual use. A 6 degrees-of-freedom tracker, hidden inside a miniature plastic camera shell, is used to control the current view. The tracker is calibrated so as to match the virtual 3D model to an imaginary scaled down model that would be standing over the floor plan perfectly aligned with the drawing. Calibration is performed by registring three points with their location on the drawing.

The user may first manipulate the virtual camera by looking at the floor plan and by quickly moving the camera to a specific rough position and orientation. This rough positioning can be performed in a fraction of a second without any ambiguity, because the floor plan provides the global context. Hence, the user is never lost and colleagues in the room may easily follow what is happening by seeing the relative position of the virtual camera with respect to the plan. Furthermore, collaborators may point directly to the floor plan, or even mark it. This mode fits well with the traditional skills and practices of design review in many disciplines. The user and others in the room may look at the 3D images to get further insight and detect design flaws not easily discernible from engineering drawings. Alternatively, the user may choose to look at the screen during detailed (relative) adjustments of the camera position and orientation. A latch mechanism was developed to temporarily freeze the view by filtering out the electronic noise or muscle vibration so as to provide a steady image for careful inspection.

The limitations of the virtual camera, as described above, stem from the rigidity of the drawing, which limits the dynamic range and orientation. For example, when viewing an entire factory, one may need a low resolution floor plan to position the view in the right section, but more precise control may be needed for reviewing a specific part or detail. Increasing the resolution of the mapping requires either ignoring the floor plan and looking only on the screen, thus loosing the benefits of the virtual camera, or switching from the global floor plan to a detailed drawings of the part and updating the calibration. This particular problem was investigated by Randy Pausch and his colleagues [7] for immersive VR situations. They have chosen to use a virtual floor plan, which in general may not offer sufficient resolution for industrial applications. A non immersive solution, where the floor plan is replaced with an electronic medium, such as the large monitor or rear projection screen may be preferable but is prohibitive in cost, unless the same screen may be conveniently used for both images without significant loss of resolution, ease of use and efficiency.

5. Conclusion

The availability of inexpensive graphics solutions, of internet-based communication and collaboration channels, and of standards for incorporating images with annotations in personal productivity tools promise to make the CAD database accessible to everyone within the corporation, to its customers, and to its suppliers for a variety of tasks: collaborative design review, 3D-based multi-media problem reports, collaborative problem solving and tracking, illustrations for marketing and documentation, online training and documentation, internet-based part purchasing and subcontracting, demonstration to customers... Some major issues must be resolved before a few toy examples of this deployment turn into productivity tools for our manufacturing industry. For example, Product Data Management protocols must be interfaced with 3D synchronous and asynchronous collaboration and with design decision support and problem tracking tools; standard data exchange formats must be fully supported by the vendors of CAD and viewing systems. These problems are exacerbated by social and economical difficulties such as code legacy, confidentiality and competitiveness issues between companies, user acceptance, and resistance to changes in the overall design process. We focus here on a few more technical aspects and discuss recent advances that address ease-of-use and performance issues for complex data sets.

Ease-of-use is addressed with the "virtual camera", developed by the author and his colleagues at IBM Research, which offers users the hand-eye coordination of immersive virtual reality without its drawbacks. It is particularly well suited for collaborative design review. With the virtual camera a novice user with no training in 3D navigation becomes fully productive in 30 seconds and can focus on the real design problems, instead of on the interface.

Data complexity found in commercial CAD databases, especially in the automotive, space, and construction industries, significantly exceeds the capabilities of any interactive graphics system and the bandwidth of internet communication channels. Geometric simplification and compression techniques developed by the author and his colleagues at IBM Research improve transmission and graphics performance by more than an order of magnitude.

Acknowledgments

The work on 3D Geometric Compression was performed at IBM research in collaboration with Gabriel Taubin. The approaches to simplification were also performed at IBM in collaboration with Paul Borrel and Josh Mittleman for the vertex clustering approach and in collaboration with Remi Ronfard for the iterative edge-collapse approach. The virtual camera was conceived by the author and developed by Bob Wolfe, in collaboration with Peter Kirchner, Jai Menon, and Paul Borrel. Product quality implementations of compression, simplification, and the virtual camera have been incorporated within IBM's 3D Interaction Accelerator.

References

[1] “The IBM 3D Interaction Accelerator”, P. Borrel, K.S. Cheng, P. Darmon, P. Kirchner, J. Lipscomb, J. Menon, J. Mittleman, J. Rossignac, B.O. Schneider, B. Wolfe. IBM Research Report RC 20302, 12/4/95,

[2] "Triangulating multiply-connected polygons: A simple, yet efficient algorithm", R. Ronfard and J. Rossignac, Proc. Eurographics'94, Oslo, Norway, Computer Graphics Forum, Vol 13, No 3, pp. C281-C292, Sept 1994.

[3]"Geometric Compression through Topological Surgery" by G. Taubin and J.Rossignac. IBM Technical Report RC-20340, 01/16/96. (www.watson.ibm.com:8080/PS/7990.ps.gz)

[4] “Multi-resolution 3D approximations for rendering complex scenes” J. Rossignac and P. Borrel, pp. 455-465, in Geometric Modeling in Computer Graphics, Springer Verlag, Eds. B. Falcidieno and T.L. Kunii, Genova, Italy, June 28-July 2, 1993.

[5] “Full-range approximations of triangulated polyhedra”, R. Ronfard and J. Rossignac. Proceedings of Eurographics'96, Computer Graphics Forum 15(3), C67-C76, 1996.

[6] “Geometric Simplification”, J. Rossignac, Lecture D in the Siggraph Course Notes #35 on Interactive walkthrough of large geometric databases, New Orleans, Aug 1996.

[7] “Navigation and Locomotion in Virtual Worlds via Flight into Hand-Held Miniatures”, R. Pausch, T. Burnette, D. Brockway, M Weiblen, Computer Graphics, Siggraph Proc., pp. 399-400. August, 1995.