Eastern Academic Journal

ISSN: 2367–738
Issue 3, pp. 46-52, September, 2018

APPROACH FOR MESH OPTIMIZATION AND 3D WEB VISUALIZATION

Pavel I. Hristov [1], Emiliyan G. Petkov [2]

[1] Pavel I. Hristov
Faculty of Mathematics and Informatics,
St. Cyril and St. Methodius University,
Veliko Turnovo, Bulgaria
pavel.hristov@outlook.com

[2] Emiliyan G. Petkov
Faculty of Mathematics and Informatics,
St. Cyril and St. Methodius University,
Veliko Turnovo, Bulgaria
epetkov@abv.bg

ABSTRACT — *This article presents techniques and solutions for publishing reality-based 3D models online. Mesh simplification method is described as a valuable solution for reducing data size and better three-dimensional web user experience. The represented approach describes a process of decreasing size of 3D file by largely simplifying flat areas thus preserving geometric features and visualizing reduced textured three-dimensional model on web. A procedure of object scanning, X3D model viewer representation, including X3D files with Inline node, used within a framework for integrating and manipulating scenes, are also described. These solutions are used for 3D exhibits' visualization on St. Cyril and St. Methodius University of Veliko Turnovo museum's web page. Main results are summarized for three-dimensional representation methods and technologies. This report also gives an overview of techniques, conclusions and various approaches applied.*

Keywords: mesh, optimization, system, scanning, segmentation, visualization, X3D

1. INTRODUCTUON

The study includes several important tasks - research of methods and new approaches for 3D visualization of geometric models on the Internet, technologies and approaches for digitizing three-dimensional objects, processing, editing, storing museum exhibits, and visualization with a high level of detail and realism.

This report presents methods for 3D scanning of a museum exhibit, mesh optimization and web visualization as part of a comprehensive scientific research for creation and development of web-based information system for visualization of three-dimensional models of museum exhibits. It leads to essence of the problem - digital storage of objects, users' expectations of such type of virtual museums, specialized spaces with similar appearance and the idea of realizing and presenting major parts of an exhibition in a digital form [5].

Approaches are described for storing objects, editing, preserving museum exhibits as 3D models, creating an information system for visualizing three-dimensional models of museum exhibits, exploring 3D technologies for web, presenting a workflow on 3D scanning of museum exhibits, and implementation via X3D standard for web visualization. Main benefit of the technologies listed is 3D visualization without need for installing additional apps and plugins in a web browser, which gives much better user and web experience.

High-resolution animated 3D models, its deforming meshes or time-varying surfaces, are widely used in various applications. These models are required in order to present details and fine structures. However, some specifics might be unnecessary from different perspectives. Mesh simplification is a process of eliminating such unnecessary or redundant details from high-resolution 3D models by removing vertices, edges and faces. When applying this process repeatedly, an animated model can be converted into a set of progressive mesh, representing a sequence of three-dimensional meshes with continuous level of details. As a result, complex models are needed to be reduced for use of mesh simplification algorithms. Mesh optimization can also be defined as a process of reducing number of faces and vertices of a given input mesh while maintaining a stable approximation to the original mesh [4].

Automatic simplification to highly detailed mesh model received high attention in research fields such as real-time rending of large-scale terrain, rapid prototyping, surface reconstruction in reverse engineering. Various kinds of optimization algorithms such as vertex or triangle decimation for removing vertices and re-triangulating surrounding mesh, vertex clustering to condense vertices in certain cell into one vertex, iterative edge and vertex pair contraction have been analyzed.

Due to detailed and accurate presentation of museum exhibits, an advanced structured DAVID 3D light scanning technology is used for processing in order to capture physical dimensions of an object, presenting and assembling individual scans in a digitalized 360-degree model.

2. SCANNING METHOD

A technological diagram is created, visualizing complete working process, geometric model preparation, individual scanning development, assembling a complete model, storing in a suitable graphical format, information system integration, and 3D model web visualization of a museum exhibit.

3D scanning and digitization can be used as a precise method for digitized transfer of physical object in an organized way, for a result of 3D scanned data. DAVID 3D platform provides digitization of reality with extreme precision. Enhanced structured light scan technology captures physical dimensions of an object and creates a 360-degree model with a resolution of up to 0.05 mm and 2.3 million vertices of scanning. Figure 1 shows an accurate scanning and modelling of individual layers, assembling all separate scans for a real-time experience.

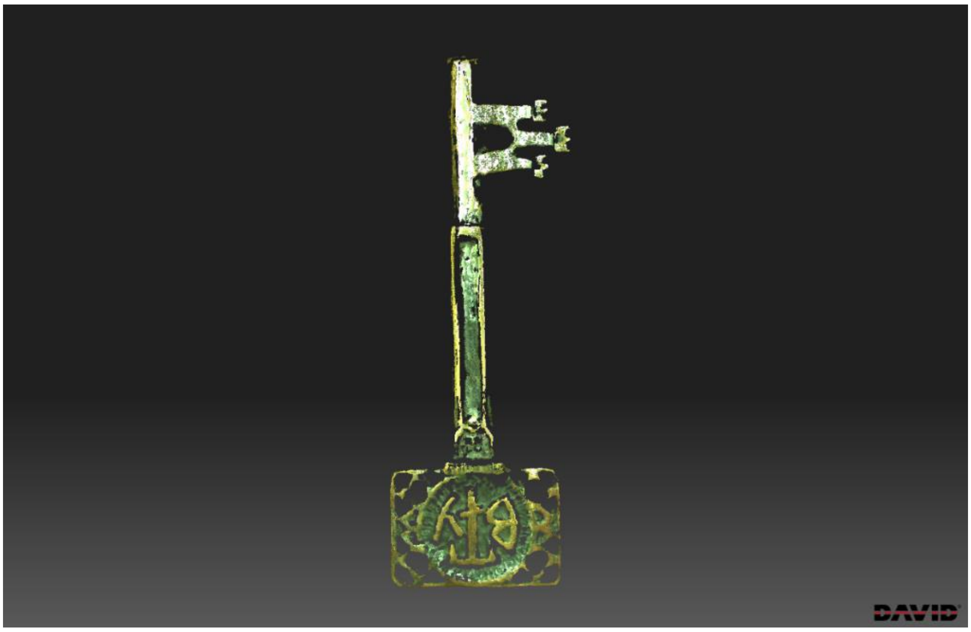

Figure 1. Scanned museum object in DAVID 3D platform.

In order to ensure that multiple scans of an object can be well combined at later stage, sufficient overlapping is a necessary process step. This requires a minimum of eight scans around the object as well as angular scans from top and bottom views. Texture can also help matching all scans in a precise manner. By filtering all results, the object can be smoothed out and adjusted for a three-dimensional scan, as well as removing unwanted background from scene around the scanned model.

Scanning software also needs to go through a calibration process in order to capture objects of varying sizes using graphic panels and achieving accurate and unobstructed 3D data collection. Exporting 3D models can be achieved through OBJ, STL and PLY file formats, compatible with a wide range of popular 3D CAD programs. All 3D scanners perform a regular sampling of the surface of an object, returning triangle meshes of complexity directly proportional to scanner's sampling resolution and object surface area. Meshes typically produced have thousands or even millions of triangles. Complex meshes present overheads in transmission, rendering, processing and storage. Hence, visualization of such models may have interference in real-time realization without mesh optimization techniques as to reduce number of rendered triangles while maintaining mesh quality. For digital geometry processing, most scanned models should carry complete re-meshing pattern before further processing. Geometry-processing algorithms concerning smoothing and compression approach can benefit from parameterization-based re-meshing methods, combined with uniform or curvature adapted sampling procedures [3].

3. APPROACH FOR 3D MESH OPTIMIZATION

Processing and editing triangular meshes, providing tools for editing, inspecting, rendering, texturing and mesh conversion, processing raw data produced by 3D digitization tools, devices and preparing models for 3D printing, are all essential tasks for interactive visualization. Certain number of polygons can be reduced, issuing less storage, faster rendering and manipulation. Overall result is generality, efficiency and scalability. Primary concern becomes bandwidth optimization, by means of minimizing storage requirements. A simplification algorithm can proceed the original highly detailed model, created by a laser scanner or CAD program and reducing it to a bandwidth-friendly level of complexity. If reducing size is required to store or transmit three-dimensional models, then algorithms for geometric compression should also be taken into consideration [10].

Analysis have been carried out for simplifying meshes of geometric 3D models. Some of the methods provide sufficient result for optimized meshes that represent refined shapes while preserving features of the original model in regards to small-scale data size. Other options simplify mesh iteratively and store withdrawn information that can be also used for progressive reconstruction of the original model with least quality issues.

Modern scanning techniques can deliver surface meshes with millions of triangles. As such highly detailed meshes are costly to process on one hand and contain a lot of redundant geometric information on the other, an optimization phase has to be established before further processing. A simplification algorithm can reduce number of triangles while preserving overall shape or other properties of a given mesh. This approach is commonly used in computer graphics to generate different versions of a three-dimensional object at various levels of detail as seen in Figure 2. It can be also noted that the symbolic cup model preserved major parts of its details in the decimation process due to quadric edge collapse decimation approach. Optimal position of simplified vertices is assured when collapsing an edge, while chosen vertex position minimizes current estimated error. Rendering efficiency can be increased by adapting object complexity to current distance between camera and object. Simplification method can further be used to store and transfer geometric data progressively [7].

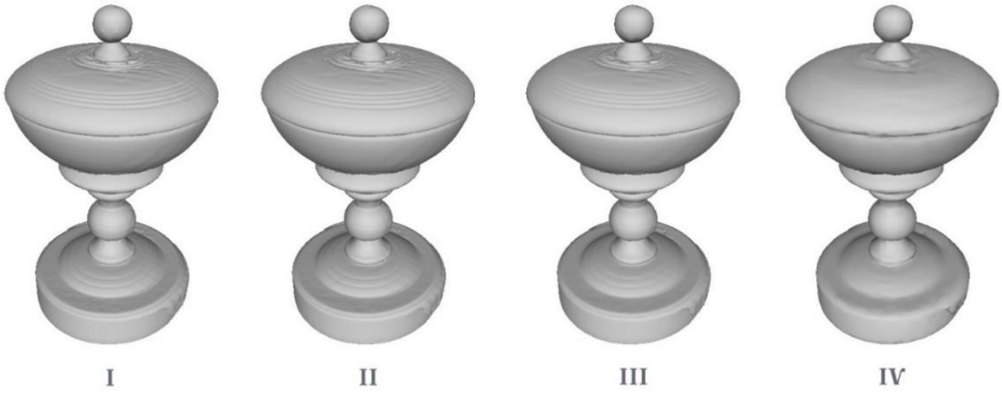

Figure 2. Symbolic cup model at different levels of detail with 739.363 (I), 339.363 (II), 89.363 (III), and 9.363 (IV) triangles.

Approximation issue of a given input mesh with less complex but geometrically conscientious representation is well-established in computer graphics. Given visual complexity required in order to create realistic-looking scene, simplification approach can be essential for efficient rendering. Level of detail representations stand positively in real-time applications such as virtual reality, scientific visualization, terrain design, representing significant demand for effective mesh simplification methods for web [11].

A practical solution can be achieved with external libraries. Computational Geometry Algorithms Library ensures an access to efficient geometric algorithms in the form of a C++ library. It offers data structures and algorithms such as triangulations, Voronoi diagrams, Boolean operations on polygons and polyhedral. On the other side OpenMesh library provides a generic data structure for representing and manipulating polygonal meshes. It can also be used with OpenFlipper, flexible geometry modeling and processing framework which presents an interface for creating and testing geometry processing algorithms [8].

Creation of 3D geometry occurs in many fields, however, type and structure of geometry created depends on application. Certain kinds of modeling techniques that are acceptable for 3D animation are not sufficient for engineering purposes. Only surfaces are needed for entirely visual 3D modeling. Edges of the surface can be wrapped on other surfaces or hidden when the model is rendered. Design and engineering applications usually require creation of surfaces’ collection where edges meet and no dents are visible. These surfaces are defined by collection of points, associated with lines and face construction, usually triangles or quadrilaterals. For renderings or animations, it is not mandatory what shape these faces appear, the normal direction has to be set correctly so it can be properly visible for rendering. In engineering applications, like finite element analysis, shape of mesh is an important matter, where sharply acute angles in faces of mesh elements cause engineering equations to produce errors and unsatisfactory results.

In Three.js, mesh elements are oriented mainly towards animation. Using GeometryUtils.merge() function, mesh objects can be combined in any number and orientation thus creating simplified and easier to load object. It is important for improving performance that number of WebGL calls be reduced. If less data is exchanged between CPU and GPU, the better will be for performance matter. THREE.GeometryUtils.merge() supports merging geometries with meshes. In this case, the function will take mesh position and orientation. If the mesh uses THREE.MeshFaceMaterial function, materials will be copied as well. This case is quite convenient when building a large geometry, or when optimizing existing scenes with meshes.

That function can be used whenever possible due to performance improvement. Merged geometry will act as a single atomic shape thus reducing the amount of WebGL calls. When merging geometry, creating a new mesh can be avoided. It can be reused, position changed, merged, and process repeated. This approach provides speed up process and saves memory [6].

There are also open-source JavaScript libraries designed with simplicity, performance and usability in mind. Leaflet is a widely used open source JavaScript library which is a base for mobile and desktop platforms, supporting HTML5 and CSS3. It can be extended with lots of plugins, has well-documented API and vector layers: polylines, polygons, circles and rectangles. This solution features smart polyline and polygon rendering with dynamic clipping and fast simplification approach. Simplify.js is a high-performance JavaScript 2D/3D polyline simplification library, extracted from Leaflet which uses a combination of iterative end-point fit algorithm for curving composition of line segments, finding a similar curve with fewer points, and Radial Distance algorithms. It is operational both on browser and server platforms. Polyline simplification dramatically reduces number of points in a polyline while retaining its shape, giving a huge performance processing boost while reducing visual noise. JavaScript Clipper library is also an useful option for simplifying polygons, in means of self-intersecting polygons and conversion to simple polygons. The result is a polygon which has self-intersecting parts, divided into multiple simple polygons.

Various methods have been analyzed which differ widely in approach, efficiency, quality and generality. Some techniques offer efficient processing but produce simplified meshes with visual issues. Other approaches create better approximations but are slow and difficult for implementation. Some algorithms lead to great lengths in order of preserving topology of input mesh. There are also methods restricted to performing well on manifold surfaces. In practice, which algorithm is best suited for performing given simplification process depends heavily on characteristics of input mesh and desired attributes of approximation process whereas techniques presented are constructive way for relatively fast simplification and visualization with less bandwidth leading to better user experience.

5. WEB VISUALIZATION RESULT

After the steps taken for mesh optimization with respect to quality criteria, the object can be exported to X3D standard with utilization of HTML5 specification for declarative three-dimensional scenes. X3D file format visualizes object surfaces by triangle subdivision or approximation and geometry storage of each vertex of the mesh. Triangles are then defined by pointers to described vertices. The result is triangle mesh representation by a shared list of vertices and triangles for storing vertices' pointers. X3D extension has useful features like multi-stage and multi-texture rendering, light and normal-map shading, VRML extension. Geometry component consists of four types of nodes: shape, geometry, property and appearance. Node visualization describes appearance properties, material and texture which advance to loading a single file on web, extensibility processing and real-time experience [1].

X3D model can be imported into X3DOM context with an Inline node. Files involve scenes, simple or complex objects with content manipulation during runtime. In order for averting name conflicts, nameSpace have to be used for ID's, as well as true values for mapDEFToID attribute. Every node of a model can still be accessed by using JavaScript function. Using inline command is practical for model loading within X3DOM context that results in empty context but not an empty page. User can still view and interact with website content while X3DOM loads model and shows status in upper left corner of its canvas.

Figure 3 shows X3D model viewer based on process of using CSS with X3D element. It allows styling 3D rendering view within an HTML element, proceeding web document creation with X3DOM scene. Developed section can also be amended with a button that allows resize of scene by setting CSS attributes using JavaScript functions. The model viewer described offers interactivity and 3D user experience presenting one of the most intriguing museum exhibits – the symbolic key of St. Cyril and St. Methodius University of Veliko Turnovo. X3D format also offers visualization of objects with less or greater geometric complexity directly within a web browser without need for additional applications or plugins.

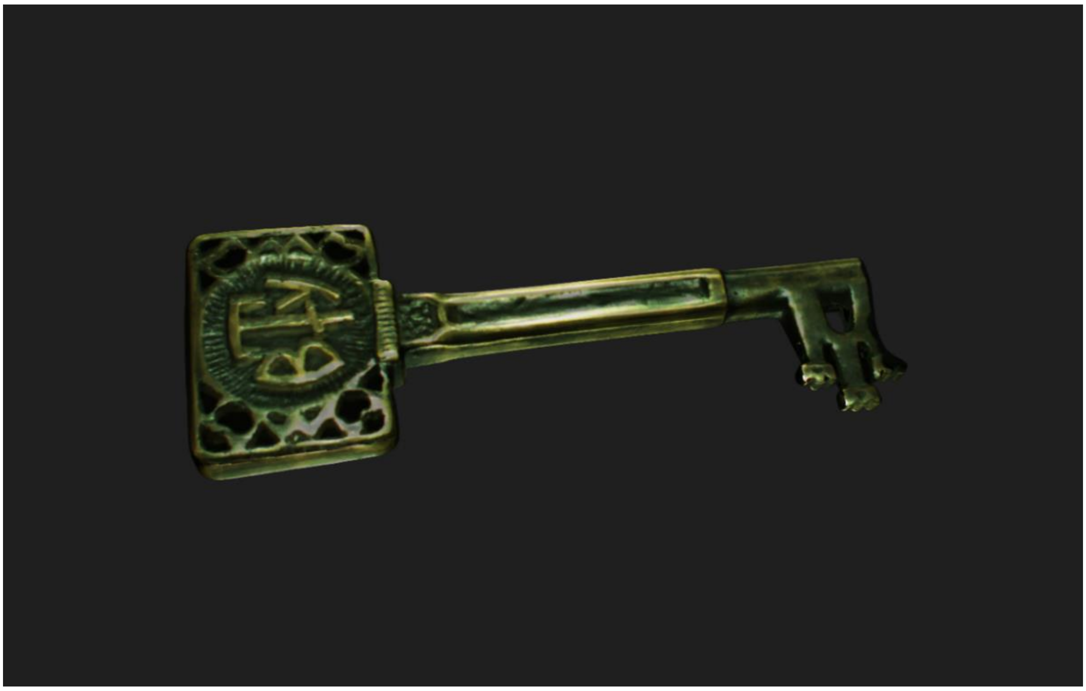

Figure 3. X3D model viewer of the symbolic key of St. Cyril and St. Methodius University.

The result can be seen rendered in a WebGL compatible browser. CSS and JavaScript code are integrated within the HTML element for faster viewer ability, resizing the scene to a certain height and width of the parent section for the body element. CSS rule is used in order to set height and width for the visual layer, ID attribute is added to the X3D element [2].

CSS background rule is also added for changing phase of the WebGL viewport. Scene needs to be transparent, in consideration of object’ proper visualization. It has to be pointed out that dimensions are relative and adapted when resizing the browser window. Dynamic extension is a valuable option for automatically adapting to browser’ resize and positioning layout elements. In regards to adding more visual interaction, full screen feature is used by changing dimensions of the X3D element.

5. CONCLUSIONS

The article describes three-dimensional representation methods and technologies, an approach of mesh decimation, leading to file size decrease for 3D object by largely simplifying flat areas thus preserving geometric features. Three-dimensional model viewer with the X3D standard is described which provides full backwards compatibility, functionally equivalent XML and compressed-binary file encodings. Three-dimensional visualization, techniques and testing methods are also summarized.

Polygon meshes can be used for simplified geometric operations and applications such as collision detection, surface analysis and permit hardware assisted rendering, as an essential condition for achieving real-time display. Three-dimensional models are widely used in many applications like virtual reality, entertainment, computer-aided design, scientific simulation and e-commerce platforms. Among various existing representations, triangular meshes are particularly appropriate for representing these models due to algebraic simplicity so that most part of manipulations can be processed by graphic hardware. The increasing popularity and size of 3D meshes to address the needs of representing objects or scenes with higher realism have become a critical issue, especially for end-users with limited bandwidth and storage capacity. In this context, compression methods have added value in this area [9].

It can also be outlined that 3D scanning and digitization can be used as a precise method for digital transfer of physical object dimensions in a detailed and organized manner while optimization of high-poly objects can be valuable for less storage and faster rendering thus maintaining material, mapping and vertex color information.

6. REFERENCES

1. Brutzman, D., Daly, L. (2007). *X3D: 3D Graphics for web authors.*
2. Gallo, G., Muzzupappa, A., Bruno, F. (2013). *3D reconstruction of small sized objects from a sequence of multi-focused images.*
3. Garland, M., Heckbert, P. (1997). *Surface simplification using quadric error metrics*.
4. Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W. (1993). *Mesh optimization.*
5. Hristov, P., Petkov, E. (2015). *Research of modern technologies and approaches for the development of Web-based information system for visualization of three-dimensional models of museum exhibits.*
6. Luebke, P., D. (2001). *A developer's survey of polygonal simplification algorithms.*
7. Nan, L. (2008). *A new adaptive mesh simplification method using vertex clustering with topology and detail preserving.*
8. Schroeder, W., Zarge, J., Lorensen, E. (1992). *Decimation of triangle meshes.*
9. Stanco, F., Battiato, S., Gallo, G. (2011). *Digital imaging for cultural heritage preservation - analysis, restoration and reconstruction of ancient artworks.*
10. Touma, C., Gotsman, C. (1998). *Triangle mesh compression.*
11. Wünsche, B. (1998). *A survey and evaluation of mesh reduction techniques.*