

Generating GIF from 3D Mesh and Point Cloud: An Automated Software

Samart Moodleah*, Pornsuree Jamsri*, Waqar Shahid†

Email: {samart, pornsuree}@it.kmitl.ac.th, waqar.shahid@gmail.com

*Faculty of Information Technology, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand

†Department of Mechatronics Engineering National University of Sciences and Technology (NUST) Islamabad, Pakistan

Abstract—We present an automated software to create a GIF animation using a sequence of images extracted from a 3D model. The software uses the features of the model such as: vertices, faces, colors, and normals. A quality of the animated GIF depends on the number of extracted images.

Keywords—GIF, Point Cloud, 3D Mesh

I. INTRODUCTION

3D CAD (computer aided design) models are used in many areas; 3D printing industry uses them as the objects that later are transformed into a collection of machine coordinates, x-y positions, which tell printers nozzle where to move, video games industry uses them to create characters, architecture sector uses them to represent their design, medical industry uses them to analyze the human organs, etc. A 3D model represents a physical body by a set of points in the 3D space, associated with a variety of geometric primitives such as lines, triangles, polygons, etc. Generating 3D models can be done manually, using mathematical modeling, scanning, reconstruction, etc.

Thousands of free and commercial 3D models are available online in multiple formats e.g. 3DS, BLEND, DAE, OBJ, STL, PLY. The popular 3D repository websites such as Thingiverse [1], GrabCAD [2], or Sketchfab [3], provide a large collection of 3D models and display them as static images for users to preview the available objects, their color, size, texture, shape, etc. However, in many cases, an image is not enough to decide whether it is what the users are looking for.

Repository sites such as Thingiverse provide static thumbnail images that allow users to see the 3D models under different angles. When the image is selected (purchased), the downloadable files are prepared in several formats.

On the other hand, Sketchfab [3]–[5] provides a strategy to explore its contents using two steps: static image thumbnail and WebGL. The WebGL technology allows the users to better visualize the object using rotations, zoom, etc

However WebGL might be affected by high internet traffic which slows down the interaction with the user. Therefore, many websites [6], [7] display 3D models using the GIF animation format that enables fast visualization. Facebook and other social media have now incorporated plugins that can show the 3D models and require gITF tool [8].

In this paper, we present an automated software to create a GIF animation of a 3D model using a sequence of images extracted using vertices, faces, colors, and normals. The

software uses open source libraries vcglib [9], imagemagick [10], OpenCV [11], and OpenGL [12]. The code is publicly available at [13]. The model has been tested on a variety of 3D models and shows reliable and visually consistent results.

II. LITERATURE REVIEW

Presenting 3D content online on a web site requires a high internet speed and a large memory compared to the other media such as text, image, video or sound [5].

The smallest element of the 3D mesh topology is three connected vertices, the plane containing the three vertices known as face, and a to the plane. The early 3D web formats are VRML (Virtual Reality Modelling Language) and its subset X3D (where the X stands for Extensibility). The 3D data is visualized by specific tools which might require user to install a plug-in in order to display the content properly. This is a considerable drawback of this approach because not only it relies on the user technical knowledge, but also requires a specific visualization tool for different web browsers.

This disadvantage has been overcome by WebGL. WebGL changes a strategy to display the 3D content online by enabling web browsers to natively render 3D data using the features of 3D graphic hardware of end-user devices without an extra plug-in. WebGL is available on multiple platforms: Windows, MacOS, iOS, Android [14]. However, WebGL requires professional computer graphic programming.

3D content can be stored using different formats. The STL format has been standard for an industry and is used in prototyping, additive manufacturing e.g. 3D printing [15]. The popularity of STL file format is its simplicity because it majorly contains only mesh information: vertices, faces, normal. The major disadvantage is a lack of color or texture information. The recent formats such as STEP, PLY, OBJ, DXS, 3DS, and SLC have both pros and cons that in some way gaining more popular and become more important format in several areas of use [15], [16]. In this paper, the Polygon File Format or the Stanford Triangle Format known as PLY and the OBJ, i.e. geometry definition file format is used to demonstrate the proposed framework.

A PLY/OBJ format was originally designed to store a data from a 3D scanner flat polygons. It is also capable of storing multiple properties such as color, surface normal, texture coordinates, etc. This format is stored as an ASCII or binary file. Figure 1 presents an ASCII version of the PLY format and Figure 2 shows a mesh topology. It consists of two major parts: a header and an element. The first part

```

ply
format ascii 1.0      { ascii/binary, format version number }
comment made by anonymous { comments are keyword specified }
comment this file is a cube
element vertex 8      { define "vertex" element, 8 in file }
property float32 x    { vertex contains float "x" coordinate }
property float32 y    { y coordinate is also a vertex property }
property float32 z    { z coordinate, too }
element face 6        { there are 6 "face" elements in the file }
property list uint8 int32 vertex_index
                    { "vertex_indices" is a list of ints }
                    { delimits the end of the header }
                    { start of vertex list }

end_header
0 0 0
0 0 1
0 1 1
0 1 0
1 0 0
1 0 1
1 1 1
1 1 0
4 0 1 2 3          { start of face list }
4 7 6 5 4
4 0 4 5 1
4 1 5 6 2
4 2 6 7 3
4 3 7 4 0

```

Fig. 1: PLY format

presents general information of an object such as number of vertices, number of faces, and other properties. The second part includes vertices coordinates, faces (vertex-indices), vertices color, texture coordinates, and other user defined properties. OBJ format is another widely used format to store 3D objects first developed by Wavefront Technologies in its advanced visualizer package (Paulbourke website [17]).

Graphic Interchange Format (GIF) is considered as one of the most popular to represent images on the web, one important feature is the ability to encode simple animations by storing a number of images in a single file used for the animation [18]. GIF is essentially an indexed file storing multiple slightly different images with a certain time-delay suitable for displaying a simple animation or a video clip.

In this paper we use a GIF animation which is a compromise between a static image and the WebGL model.

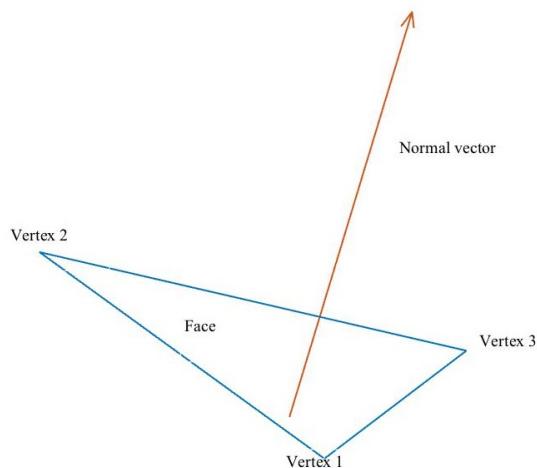


Fig. 2: Triangular mesh

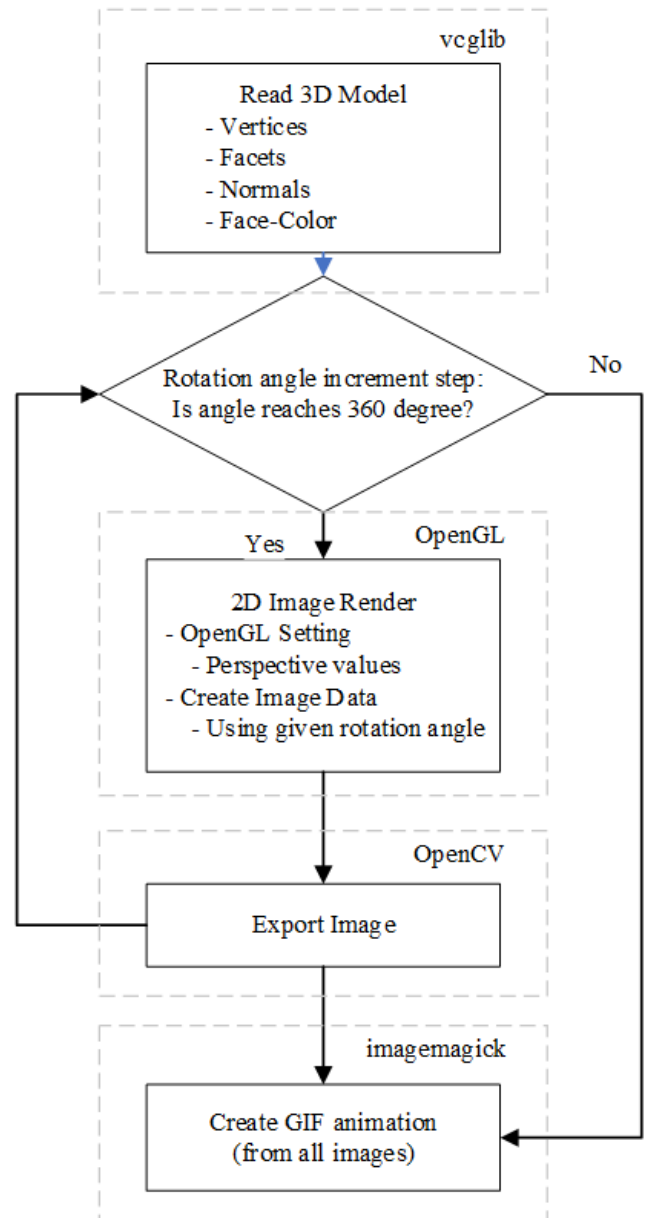


Fig. 3: Framework diagram

III. METHODOLOGY

Our novelty lies in proposing the software framework that automatically generates a GIF animation from a 3D model, converting from 3D to 2D space. A framework diagram is illustrated in Figure 3, it comprises of four steps: (1) reading a 3D model, (2) rendering the images, (3) exporting the images, and (4) creating a GIF animation. The details of each parts and the open source libraries used in paper are explained in this section.

The first step, a 3D model is read and kept in the vcglib data structure. The result can be concluded into two scenarios: with and without a texture image (depends on an input 3D model). In the first scenario, a texture image is used for texture

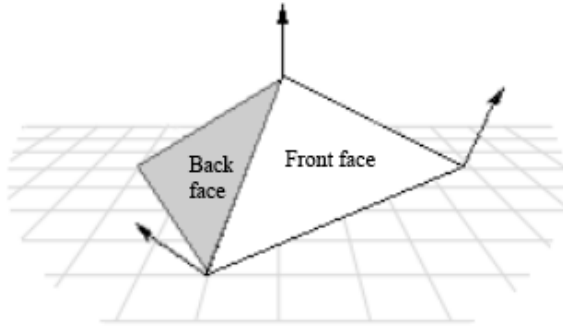


Fig. 4: A front (visible) and a back face of a polygon

mapping. The second scenario, a 3D model provides only the vertices color which is later used to approximate the faces color values. Either the first or second scenario, all faces-color is determined.

Note that each face of a mesh has a perpendicular unit normal vector; its direction is defined by the order in which the vertices are determined and by whether the coordinate system is left- or right-handed. In computer graphics, the unit normal vector is used to define Gouraud shading, lighting, and texturing effects [19]. The face normal pointing away from the front side of the face is visible (front face) while the back face (normal vector points in the opposite direction) is not rendered. Figure 4 shows an example of a face normal vector.

In order to make a set of images slightly changing in time, a 3D object is rotated with predefined angles, the rotation angle is continuously incremented until it reaches 360 degrees.

The second step is a collection of 3D data is used for image rendering using OpenGL library. It starts with evaluating an approximate size of the object, setting the perspective values in order to project a 3D data onto 2D space.

At the third step, the 2D projection is exported as an image file (e.g. jpeg) using OpenCV library. This process is repeated until the 3D model is fully rotated (see Figure 5).

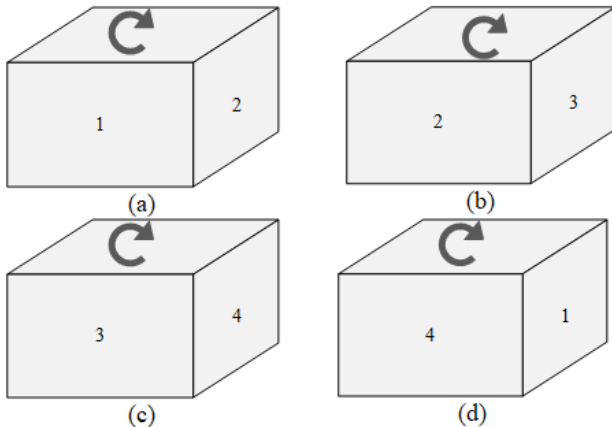


Fig. 5: Object rotation

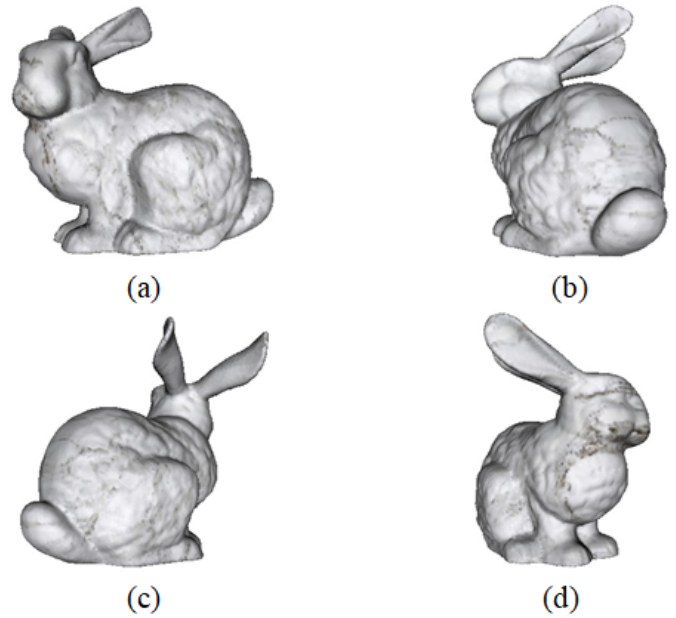


Fig. 6: Images of Stanford bunny for different rotation angles

At the last step the images are combined to generate a final GIF.

IV. RESULT AND DISCUSSION

The software is an open source written in C or C++. We tested on Linux operating system (Ubuntu) with Intel Core i7-5500 2.40GHz and 8GB of Ram. The code is available on github [13]. For software evaluation, we choose seven 3D models generated using OpenMVG [20] and OpenMVS [21]. Depending on the resolution of images, the size of the 3D Model varies.

An object is rotated around Y-axis (or any other axis) with a certain step until it reaches 360 degrees and finally yields a sequence of images for GIF generation. We evaluate the proposed framework based on two conditions: a fixed size of images (500x500 pixels) and the 3 incremental steps of rotation angles (by 5, 15, and 45 degrees).

The four rabbit images are obtained with the step 90 degrees are shown in Figure 6 and the two frames generated from each GIF are shown in Figure 7 for another 6 objects. The qualitative analysis is given online [22].

The results displayed in Figure 8(a) and Figure 8(b) represent a framework to reduce the total processing time and the size of the animation file size. Note that the number 5, 15, and 45 in Figure 8 represent the incremental steps of rotation angle.

A quality of animated GIF depends on many factors, e.g. a number of images that reflects the object smooth transition. For example, 15 degrees; a medium quality in this test, a Bunny (69451 faces) requires 3.5 seconds to process and 1.1 MB storage. The users are able to trade between quality and size of the output.

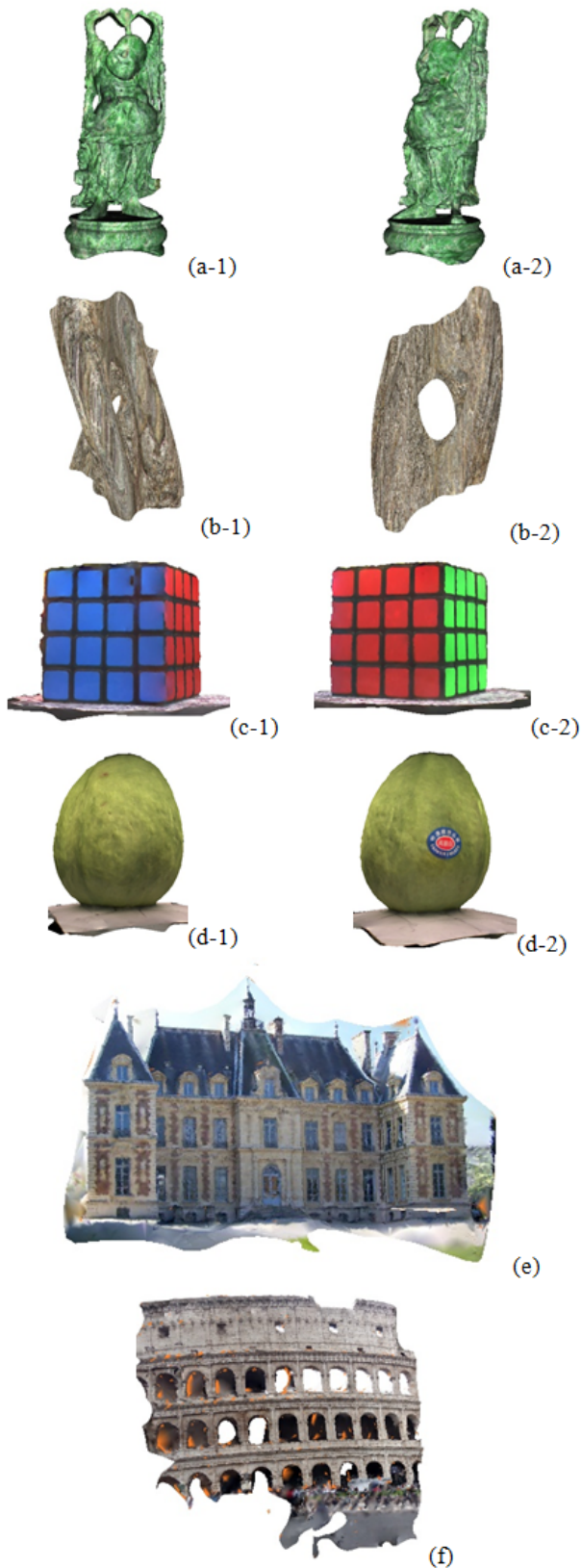


Fig. 7: Framework example of output images to make GIF: (a) Buddha, (b) Vase, (c) Cube, (d) pear, (e) House and (f) Museum

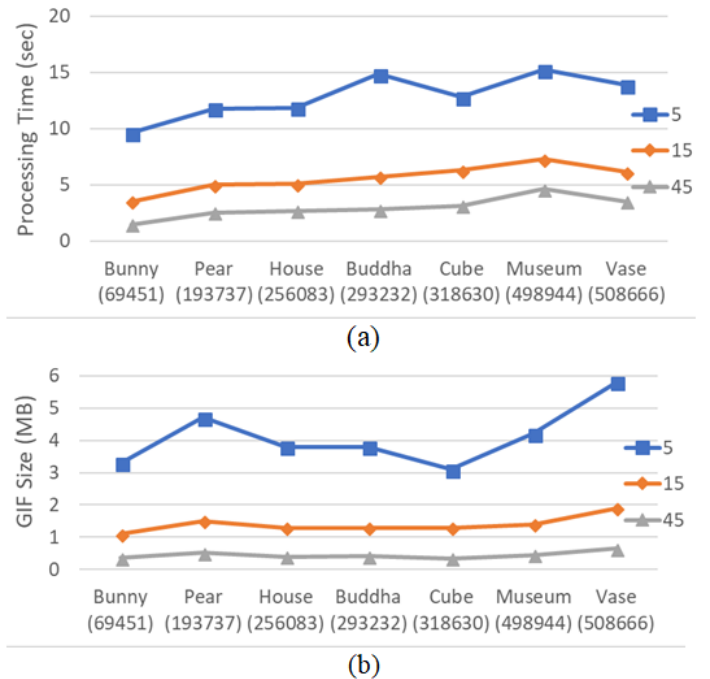


Fig. 8: Framework performance (a) processing time and (b) output file size

V. CONCLUSION

In this paper we proposed an automated framework to generate an animated GIF to view 3D models. We tested the algorithm on seven textured 3D models using view angles and different file size. The proposed software successfully generates the GIF files suitable for inspecting 3D models without applying any specific geometric software. The software is available publicly at [13].

REFERENCES

- [1] <https://www.thingiverse.com/>, (Accessed on 08/09/2018).
- [2] <https://www.grabcad.com/>, (Accessed on 08/09/2018).
- [3] <https://sketchfab.com/>, (Accessed on 08/09/2018).
- [4] S. Villagrasa, D. Fonseca, E. Redondo, and J. Duran, "Teaching case of gamification and visual technologies for education," *Journal of Cases on Information Technology (JCIT)*, vol. 16, no. 4, pp. 38–57, 2014.
- [5] R. Scopigno, M. Callieri, M. Dellepiane, F. Ponchio, and M. Potenzi, "Delivering and using 3d models on the web: are we ready?" in *Virtual Archaeology Review*, vol. 8, no. 17. Universitat Politècnica de València, 2017, pp. 1–9.
- [6] "Turn 3d models into interactive 360 spins with sirv's 360 degree viewer," <https://sirv.com/demos/360-spin-3d-model/>, (Accessed on 08/09/2018).
- [7] "Animated gif publishing software," <https://www.webrotate360.com/products/webrotate-360-product-viewer/create-animated-gif.aspx>, (Accessed on 08/09/2018).
- [8] "Richer 3d posts on facebook and new ways to share - facebook for developers," <https://developers.facebook.com/blog/post/2018/02/20/3d-posts-facebook/>, (Accessed on 08/09/2018).
- [9] <http://vcg.isti.cnr.it/vcglib/>, (Accessed on 08/09/2018).
- [10] <https://www.imagemagick.org/script/index.php>, (Accessed on 08/09/2018).
- [11] <https://opencv.org/>, (Accessed on 08/09/2018).
- [12] <https://www.opengl.org/>, (Accessed on 08/09/2018).

- [13] W. Shahid and S. Moodleah, "3d2gif," <https://github.com/waqarsqureshi/3d2gif>, (Accessed on 08/09/2018).
- [14] P. Cozzi, *WebGL insights*. AK Peters/CRC Press, 2015.
- [15] J. D. Hiller and H. Lipson, "Stl 2.0: a proposal for a universal multi-material additive manufacturing file format," in *Proceedings of the Solid Freeform Fabrication Symposium*, no. 1. Citeseer, 2009, pp. 266–278.
- [16] K. McHenry and P. Bajcsy, "An overview of 3d data content, file formats and viewers," *National Center for Supercomputing Applications*, vol. 1205, p. 22, 2008.
- [17] "Object files," <http://paulbourke.net/dataformats/obj/>, (Accessed on 08/09/2018).
- [18] W. Burger, M. J. Burge, M. J. Burge, and M. J. Burge, *Principles of digital image processing*. Springer, 2009.
- [19] "Face and vertex normal vectors (direct3d 9) — microsoft docs," <https://docs.microsoft.com/en-us/windows/desktop/direct3d9/face-and-vertex-normal-vectors>, (Accessed on 08/09/2018).
- [20] "openmvg/openmvg: open multiple view geometry library. basis for 3d computer vision and structure from motion." <https://github.com/openMVG/openMVG>, (Accessed on 08/09/2018).
- [21] "cdcseacave/openmvs: open multi-view stereo reconstruction library," <https://github.com/cdcseacave/openMVS>, (Accessed on 08/09/2018).
- [22] W. S. Qureshi, "Gif animation results," <https://sites.google.com/alumni.ait.asia/waqarsqureshi/home/3dv-data>, (Accessed on 08/09/2018).