Methodology to create optimized 3D models using Blender for Android devices

Pooja Bhawar, Nitin Ayer, Sameer Sahasrabudhe
Department of Computer Science and Engineering,
IIT Bombay, Mumbai, India
e-mail: {bhawarpooja4, ayernitin, iamsameerss}@gmail.com

Abstract — Use of Low Cost Access Devices (LCADs) based on Android platform is on the rise for educational purposes. There is a growing need for open source educational content in order to tap the advantages of this platform. 2D/3D animation is preferred to explain complex visualization concepts. Most of the 3D animation content available for desktops has high-poly models. To use these models on LCADs is challenging, because of their low hardware configuration. Methodologies to port 3D content to LCADs (used in gaming industry) are unsuitable for educational visualizations, as they hamper functionality and mesh topology of the objects. It is desirable to remodel 3D content with an objective to use them for LCADs.

In this paper we present one such methodology to create low-poly 3D models using open source 3D tool, Blender. The models created using this methodology show reduction in file sizes, mesh-density and loading time on LCADs.

Keywords- optimization of 3D models, open source 3D animation, Blender 3D, Aakash tablets, Android based LCAD

I. Introduction

Usage of Android based Low Cost Access Devices (LCADs) is on the rise in Indian subcontinent, mainly because of portability, cost and ease of access [1]. Government of India is tapping the potential of such technologies through the 'National Mission on Education through Information and Communication Technology' (NME-ICT) [2], [3]. One of the aims of NME-ICT is to provide interactive knowledge resources on a LCAD titled, 'Aakash' [4]. Aakash offers necessary functionality for educational purposes, at an affordable cost [4]. The mission proposes to maximize usage and reach of open source educational content using Aakash.

Pilot projects of NME-ICT include video lecture repository [5], virtual labs [6], and learning object (LO) repository (Project OSCAR) [7]. LOs in this repository are mainly two dimensional (2D). However, three dimensional (3D) animations are increasingly used to explain concepts with complex visualization [8]. Project OSCAR has a repository of 3D content for various educational domains.

We have chosen this repository for our study since, it comprises of open source high resolution (high-poly) 3D models using Blender. Blender is an open source suite to create 3D content [9]. Additionally, Blender also has a game engine (BGE) to create interactive 3D content. Comparative tools are proprietary and therefore Blender is more suitable for creation of 3D educational content.

To import 3D content for LCADs is a challenge, since LCADs have low hardware configuration (see the comparison given in Table 1). Automation tools convert high-poly 3D models to low-poly, and reduce the file size. Although, popular in gaming industry, these tools fail in the education context, due to the lack of precision. We show one such example in Section III. It is important to create low-poly models which should be displayed smoothly on LCADs. We modify the standard 3D modeling methodology [10], and design a new methodology.

In this paper, we present this methodology to create functional and low-poly 3D models which retain mesh topology. We show that this methodology reduces mesh-density, file size and loading time on the LCADs.

II. Problem definition

Desktop computers have higher configuration as compared to LCADs like Aakash (see Table 1). Higher configuration is necesasry for fast and correct display of high-poly 3D models.

Table 1: Configurations of desktop computer and LCAD (Aakash)

	Desktop computer	**LCAD** (Aakash)
Processor speed	3 GHz Quad core	1 GHz
RAM	2048 MB	512 MB
Graphic accelerator	Built-in	Absent

In order to reuse available 3D content (from Project OSCAR), we import these high-poly models to Aakash. We use Blender player for Android for this import [11]. We observe that there is significant delay to load these high-poly models on Aakash (see Table 2). The loading time of these models on desktops is negligible (~1 sec); since, desktops have more processing speed, RAM and graphic accelerator to display content. On the other hand, Aakash takes more time to load, owing to low hardware configuration as compared to desktops. Additionally, there is a lack of instant and smooth display of high-poly 3D content on Aakash.

Table 2: Loading time of high-poly models on Aakash and desktops

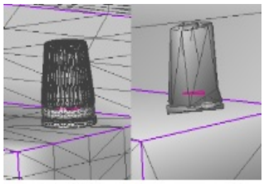
		Mesh-density *(Polygon)*	***File size*** *(Kilobytes)*	***Load Time*** *(Seconds)*	
				Aakash	***Desktop***
1	Battery	8397	672	5	Instant loading (~1)
2	Jockey	4032	650	4	
3	Ammeter	3817	989	Did not load	
4	Meter Bridge	22710	3072	10	
5	Resistance Box	18824	2201	8	
6	One way key	7668	958	3	

The mesh-density plays an important role. Models with more number of vertices and polygons, take more time to load e.g. Meter Bridge. Few models could not load, e.g. Ammeter. Apart from wastage of time, this may increase frustration and hamper the interest levels of the students.

It is therefore, important to have low-poly versions of the 3D models which will load quickly on the LCADs. It is also important to retain clean mesh topology of the models. Clean mesh topology refers to 'efficient polygon distribution, proper placement of polygonal edge-loops, few or no triangular faces (as opposed to 4-sided "quads"), and clean precise creases that minimize stretching and distortion' [12]. Absence of clean mesh topology will hinder application of materials and textures, and the final output will have unfinished look. Two ways to achieve the goals mentioned above, are: either use existing optimization tools or to remodel them. In the next section, we show the results of an optimization tool, and analyze the errors occurred.

III. Trying various optimization tools

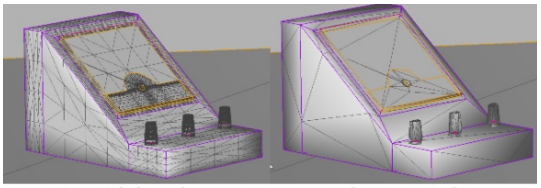
Out of the various mesh optimization tools, the polygon reduction algorithm [13] creates an optimized model (low-poly) which looks similar to the original (high-poly) model. This algorithm reduces model's complexity by repeated use of the edge (an element of polygon) collapse operation. Decimation of Triangle Meshes algorithm [14] optimizes the model by reducing the number of triangles (another element of polygon) which are required to model a particular object. It removes vertices, by reviewing triangular geometry of the model multiple times and uses local topology. However, these algorithms don't have any tools/software available. Therefore, we examine available tools to optimize 3D models. These are: Simplygon [15], Okino Polygon Reduction System [16], and Atangeo Balancer [17]. Typically, these tools are proprietary, and offer very few features for free usage. We tested one of these tools (Atangeo Balancer-trial version) to optimize existing high-poly ammeter model. Results show that the number of vertices is certainly reduced from 3817 to 767. Mesh topology is distorted and has improper placement of polygons and irregular triangulation (see the mesh shown in Figure 1b).



1a: High-poly 1b: Low-poly

Figure 1a and 1b: Mesh optimization done using Atangeo balancer

Same can be seen in the overall model of ammeter (see Figure 2b). Mesh-density is reduced by removing polygons and reducing triangular mesh, however, mesh topology is not clean.



2a: High-poly 2b: Low-poly

Figure 2a and 2b: Triangulation done by Atangeo:
Vertices are reduced, however leaves unclean mesh topology

This attempt suggests that optimizing the model without retaining clean mesh topology, is not desirable. Additionally, these tools are proprietary. Optimization process created in Blender would be beneficial in larger impact and reach. Therefore, a methodology which will retain clean mesh topology of models is necessary. In the next section, we present our methodology for the same.

IV. Our methodology To Optimize 3D models

In this section, we explain steps to be performed to create optimized 3D model, especially for creating educational content. We use ammeter and battery models, to illustrate the steps (shown using *italics*). We also explain the process administered to gather feedback about our methodology.

1. Select an object to be modeled: *Ammeter and battery*
2. List the components (parts of the model) present in the model: *Pointer, pointer fixing plate, surface mount, panel mount, connecting terminals.*
3. Classify the components on the basis of:
   a. Functional (required in the actual working of the model): *Pointer, connecting terminals.*
   b. Non-functional (required for aesthetic/decorative aspect; however, not required for the actual working of the model): *Surface mount, panel mount, pointer adjustment screw etc.*
4. For the functional components (type a), follow the standard modeling process (step 6-15).
5. For the non-functional components of (type b), perform the following steps:
   a. Decide whether specific details of the component needs to be shown or could be removed completely. *Non-functional nut-bolts in a gadget can be deleted.*

b. If the details are important and need to be shown, decide whether they should be modeled or can be shown using a texture.
Battery: The slits on the sides ventilate hot air. It is a non-functional component; however they are important from the look-and-feel point of view. These slits can be modeled (see Figure 3a); however the mesh-density would increase. Use of image textures instead of modeling retains the look and avoids increase in mesh-density (see Figure 3b).

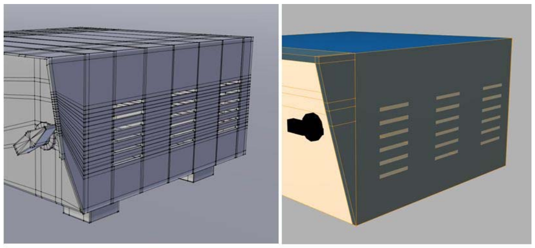

3a: High Poly 3b: Low Poly

Figure 3: Use of image textures to reduce mesh-density

c. For the details which need to be shown, but cannot be shown using textures, model them to get the 3D look. Follow the standard modeling process (step 6 - 14).
Battery: The hood, which covers the buttons is non-functional, and cannot be shown using a texture (see Figure 3). Therefore it needs to be modeled to provide the 3D look of the object.

Process for 3D modeling of individual components:

6. Select the component to be modeled, and open a new file in Blender 3D.
We have selected two components: (i) Connecting terminal and (ii) Base mount.
7. Identify a basic geometrical shape which matches the overall shape of the selected component. (Observe the contours of the object).

i. Connecting terminal: Circle.
ii. Base mount: Plane.

8. Use number of vertices, to create the look of the selected component. A circle can be represented by 12 or 64 vertices. If the circle with 12 vertices justifies the requirement, don't add more vertices.

i. Connecting terminal: Use circle with 8 vertices instead of 32 default vertices.
ii. Base mount: Start with 4 default vertices.

9. Within the component, if you find details like: (a) smooth edges, (b) shape modifications, or (c) aesthetical additions, avoid these during modeling.

i. Connecting terminal: Grooves on the terminals (see Figure 4a) provide better grip to the user, however, it is not mandatory (at least for the 3D model). Octagonal shape of the terminal (using 8 vertices) can provide same functionality as compared to 32 or more vertices (see Figure 4b). Additionally, the grooving can be used in the texture for realistic effect.

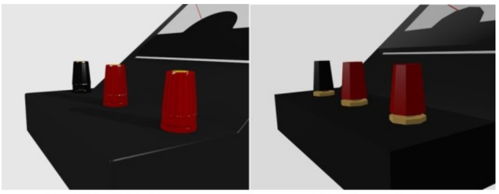

4a: High Poly 4b: Low Poly

Figure 4: Use of octagon instead of circle to reduce mesh-density and retain clean mesh topology

ii. Base mount: Actual ammeter model has rounded edges (see Figure 5a) which provides smooth contours; however, these do not add ANY functional aspect to the 3D model. Making a 3D model with these edges requires more loop cuts (and eventually more vertices). Therefore, a decision is taken to completely avoid the smooth edges (see Figure 5b). The conversion of rounded edges to basic shapes like a cube/cuboid has decreased overall mesh-density from 992 vertices to 20 vertices in low-poly model.

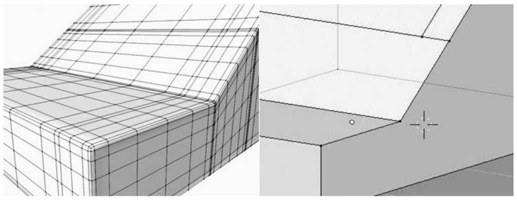

5a: High Poly 5b: Low Poly

Figure 5: Avoid unwanted smoother edges to reduce mesh-density and retain clean mesh topology

10. Proceed with 3D modeling. Use methods like scale, rotate, translate and extrude to get the basic shape of the component. Avoid using modifiers since Blender Game Engine (BGE) does not support them.

i. Connecting terminal: Use extrude and scale to complete the modeling.
ii. Base mount: Temporary application of 'subdivision surface' modifier results in 498 vertices (see Figure 6b) whereas the permanent apply setting, which has high mesh-density increases it to 31, 746 vertices (see Figure 6a).

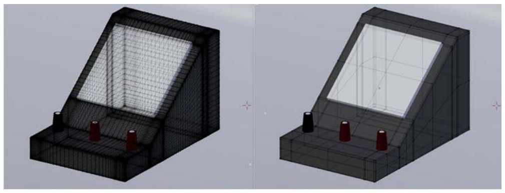
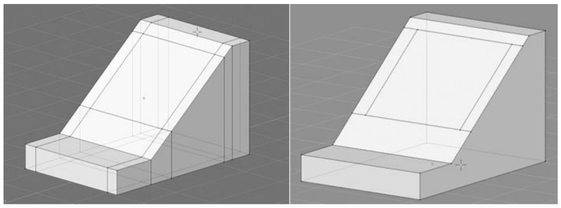

6a: High Poly 6b: Low Poly

Figure 6: Avoid use of modifiers (subdivision surface) to reduce mesh-density and retain clean mesh topology

11. **Checkpoint:** Analyze the model to remove vertices which create irregular sections do not contribute to the mesh topology. Save the file.

12. Use ‘N polygon’ method to remove extra edges (which in turn also addresses extra vertices) to decrease mesh density.

ii. Base mount: Remove extra vertices created on the faces (except front) since they are not required. Use of ‘N polygon’, reduces the vertex count from 48 to 31.



7a: High Poly 7b: Low Poly

Figure 7: Use of ‘N polygon’

13. Switch to Blender Game Engine (BGE) tab and:
    a. align the normals to face the camera.
    b. change the shade type to texture mode.
    c. change display shading properties to GLSL.
    d. apply respective materials and textures.
    e. add the required lamps to illuminate the object
15. Save the file in the default format of Blender (.blend).
16. Repeat steps 6-15 for other components in the object, until ALL the components are modeled.
17. Repeat steps 1-16 for the next model, till ALL objects are modeled.

We use our methodology, to create low-poly models of the same six objects mentioned in Table 2. Further, we test whether these models address the goals mentioned in Section II. For this, (i) we gather statistics similar to Table 2 and (ii) administer a survey with ten science teachers from nearby schools to obtain feedback about the mesh topology. Participating teachers choose image having clean mesh between low-poly or high-poly images. Next section provides the results of these tests.

V. Results

(i) Comparison of the load time between low-poly and high-poly models shows 38% reduction in file size, 91% in mesh-density and 66% in load time on Aakash (see Table 3).

Table 3: Comparison of high-poly and low-poly models about mesh density, file size and loading time on Aakash

		Mesh-density *(Polygon)*		***File size*** *(Kilobytes)*		***Load Time*** *(Seconds)*	
		L	**H**	**L**	**H**	**L**	**H**
1.	Battery	297	8397	596	672	2	5
2.	Jockey	241	4032	552	650	1	4
3.	Ammeter	204	3817	650	989	5	Did not load
4.	Meter Bridge	1558	22710	920	3072	3	10
5.	Resistance Box	468	18824	799	2201	3	8
6.	One way key	426	7668	616	958	1	3

(ii) Feedback of the teachers (n=10), shows that 89% agree that low-poly models retain clean mesh topology, and the functionality of the models is similar as high-poly models.

VI. Discussion and Analysis

The reduction in mesh-density, file size the load time is achieved by avoiding use of modifiers, loop cuts, and substituting non-functional components with image textures, wherever required. The survey results reveal that our methodology retains clean mesh topology. The teachers feel confident that lesser load time of the (low-poly) models will benefit in retaining the interest levels of the students.

Results suggest that our methodology would be more suitable for objects based on quadrilateral shapes. The methodology may not yield similar reductions, when applied to circular objects. We intend to create an automated tool based on our methodology to facilitate import of high-poly 3D content to LCADs.

VII. Acknowledgements

We would like to acknowledge the support of Project OSCAR, Aakash project and Blender group at IITB. We also thank the teachers at Campus school and KVIIT, Powai for participating in the survey.

VIII. Bibliography

[1] S. Kaur, "The Revolution of Tablet Computers and Apps: A Look At Emerging Trends," Consumer Electronics Magazine, Vol. 2-1, Jan. 2013, pp. 36-41.

[2] http://mhrd.gov.in/sites/upload_files/mhrd/files/MissionDocument.pdf

[3] http://www.sakshat.ac.in/

[4] M. Singh and P. Sethi, "Aakash: World's cheapest tablet launched; to be sold for $60 in retail," Apeejay Stya Education Research Foundation (ASERF) E News Bulletin on education, Oct. 2009, Bulletin 117, pp. 7-8.

[5] http://nptel.iitm.ac.in/

[6] http://iitb.vlab.co.in/

[7] http://oscar.iitb.ac.in

[8] T. Ellis, “Animating to improve learning: a model for studying multimedia effectiveness,” In Frontiers in Education Conference,. IEEE. Oct. 2001, Vol. 2, pp. F2G-1.

[9] http://www.blender.org

[10] S. Dere, S. Sahasrabudhe, and S. Iyer, "Creating open source repository of 3D models of laboratory equipments using Blender," Technology for education (T4E 2010), IEEE, Jul. 2010, pp. 149-156.

[11] http://wiki.blender.org/index.php/Doc:2.6/Manual/Introduction/Installing_Blender/Android

[12] http://3d.about.com/od/Glossary-T/g/Topology.htm

[13] S. Melax, "A simple, fast, and effective polygon reduction algorithm," Game Developer, Nov. 1998, pp. 44-49.

[14] W. Schroeder, J. Zarge, and W. Lorensen, "Decimation of triangle meshes," ACM Siggraph Computer Graphics, ACM, Jul. 1992, pp. 65-70.

[15] http://www.simplygon.com/

[16] http://www.okino.com/conv/polygon_reduction/geoman2/polygon_reduction.htm

[17] http://www.atangeo.com