The Analysis and Comparison of File Formats for the Construction of iOS OS Three-dimensional Objects for Augmented Reality Systems

Dmytro Ostrovka [1] [0000-0003-4818-3822], Taras Teslyuk [1] [0000-0001-6585-3715]

Peter Veselý [2] [0000-0002-7857-6355], Ihor Prots'ko [1] [0000-0002-3514-9265]

[1]Lviv Polytechnic National University, Lviv 79013, Ukraine
ostrovdi@gmai.com
[2]Comenius University in Bratislava, Odbojárov 10, Bratislava, Slovak Republic
peter.vesely@fm.uniba.sk

Abstract. The rapid development of three-dimensional technology in the modern world has also affected smart phones. An integral part of augmented or virtual reality are the file formats for displaying three-dimensional objects. Since there are more than a hundred of them, it is quite difficult to choose the most suitable for concrete project. That is why in this topic most popular files for displaying three-dimensional files have been chosen. In this work was analyzed their properties and the stack of supported technologies, their ability to use it on the iOS mobile platform, and also tested the opening speed of a particular format. During this work, an iOS mobile application was created to demonstrate how to display three-dimensional models of compared formats. The result of this work are the conclusions about the advantages and disadvantages of the compared file formats, as well as the feasibility of using one or another compared format to display three-dimensional models on the iOS mobile platform. These conclusions can be useful to developers of mobile applications for the iOS platform, which are faced with the task of using three-dimensional files in their projects. They may help save time on choosing the most suitable format, and the developed test application can serve as an example of the reflection of a file in a mobile application.

Keywords: iOS, Fbx, USDZ, USDC, USD, Collada, Obj, Augmented Reality, Virtual Reality, Mobile Development, AR, VR, Three-dimensional Files, 3d Files.

Introduction

The role of smartphones in daily lives of modern people is increasing more every day. At the same time, the market's share of sales of Apple smartphones, which operate exclusively on the iOS mobile platform, is 14.1% in the 1st quarter of 2019, which is more than 54 million devices.

Hence, the relevance of three-dimensional objects in the life of modern people is higher than ever. The development of modern technologies is directed to bring the computer world closer to the real one. Three-dimensional cinema or training with the help of virtual reality is currently, no longer new.

All these trends have led to the active integration of three-dimensional objects into such integral things for people as smartphones.

To work with three-dimensional graphics, which is quite complex, you need not only high-quality hardware, but also an appropriate software base, a certain data structure to store three-dimensional images.

Despite the high capacity of these technologies and the great prospect, there are many solutions for storing and presenting three-dimensional graphics on the market. However, some of them have restrictions in accordance with various platforms, they are not available for developers, or even outdated or irrelevant. All this makes it difficult to interact with them and also requires developers to select the format, necessary for their task and corresponding to the trends for displaying three-dimensional files.

The purpose of the article is to analyze and compare the file formats for building three-dimensional iOS OS objects. Based on this comparison, iOS OS developers will be able to choose a format for displaying three-dimensional files, based on new trends, compatibility with other formats, openness, the availability of official documentation, and iOS-based user interface. All this will help reduce development time for searching and studying all aspects of working with formats such as Obj [1], Fbx [2], Collada [3] and USDZ [4]. The results obtained can be used to develop augmented systems [5, 6] and virtual [7, 8] realities, in the process of developing smart home systems [9, 10] and smart cities [11, 12] medical applications [13, 14], devices of special [15] purpose etc.

1 Analysis of formats for saving three-dimensional objects

Due to its prevalence and widespread use in various industries, there are many file formats for building and displaying three-dimensional files. All of them have their advantages and disadvantages, the sphere of use, as well as the list of supported operating systems. Taking into consideration this, widespread formats were chosen for building three-dimensional files that are suitable for use on the iOS mobile operating system.

First, consider the **Obj** format. This is a geometry description file format first created by Wavefront Technologies for their Advanced Visualizer animation package [1]. This is an open file format, and it has been accepted by other developers of 3D editors as standard. The full name of this format type is “Wavefront Object”.

The Obj format defines only the object’s geometry, the coordinates of each vertex, its texture coordinates, normals and faces, which are defined by lists of polygon vertices. The vertices of the polygons are, by default, set counterclockwise, making explicit definition of normals to be optional.

The advantages of this format include the ease of structuring data relative to other compared formats. Also, the relevance of its use, is the support by many software tools for creating three-dimensional models. However, the relative simplicity and small file size are made up for the lack of animation, scene information, and the possibility of layering textures.

Next, consider the **Fbx** (Filmbox) format. This technology and file format were developed by Kaydara and is currently owned and continues to be developed by Autodesk [2]. This format is used to ensure the compatibility of various three-dimensional graphics programs. Fbx is a proprietary undocumented format. Today, there are two file formats of Fbx: binary and ASCI.

The advantages of the file include support for animation, scene information and texture layering. That is why the format is widely known in the film industry and video games. Also, this format makes it possible to optimize the costs of developing models with Autodesk3D animation software such as: 3DS Max, Maya 3D, Maya LT and others.

However, the main drawback of this format is its closed nature and lack of official documentation. Thus, for working with Fbx files, Autodesk forces developers to use its own Fbx-Sdk library, which makes it possible to read information about a three-dimensional model from an Fbx file.

Next, consider the **Collada** format (COLLAborative Design Activity). It is led by a consortium of nonprofit technologies, the Khronos group [3].

Information about a three-dimensional object in this format is set using the XML markup language; it increases the file size compared to the binary representation of other formats.

The advantage of this format is the support of many functions, such as animation, texture layering, the ability to save information about the scene. Also, a big benefit is the availability of official documentation.

The main drawback of the format is the absence of common standards for creating models. Together with the openness of the format, this causes a different approach to data structuring and possible conflicts / artifacts when displaying them.

Last in this article, consider the **USDZ** format. This is a file format for storing information about three-dimensional files, developed by Apple and Pixar [4]. In fact, this is an archive of USD files and their resources, such as: textures and files embedded in the same type of USDZ.

USD (Universal Scene Description) - this is the first publicly available software that solves the need for reliable and scalable, exchange and complement arbitrary 3D scenes, which can consist of many elementary resources. [19]. Pixar uses USD as the core of pipeline graphics in all of its 3D rendering applications.

USD files can be stored both in ASCII and in binary representation. Pixar positions this format as a convenient language for creating and editing 3D data, and facilitates the process of creating digital content.

The format is rapidly gaining popularity due to usability and good support. The significant factor of interest in the format, among developers and designers, is that the format is accompanied by the leading company for creating animated films - Pixar. There is also an official set of tools, USDZ Toolset, for creating, editing and converting from other formats to the USDZ format.

This format allows you to structure information in such a way as to redefine individual properties of primitives without creating a separate copy. This allows you to reduce the file size compared to other formats and also to process the file faster. Thanks to the support of nesting and links, files of this format are capable of scaling and can also be divided into separate sub-files. This allows several designers to work on the final model at the same time.

2 Comparison of file formats for displaying three-dimensional objects

Since there is no single standard that would indicate the best way to describe the file format for displaying the 3D model as a whole and specifically for the iOS platform, it is difficult to make comparisons objectively. Therefore, first of all, you need to pay attention to the requirements set in the project and the inherent functionality that has to be present.

At the same time, the following research focuses on, criteria for evaluating various formats, namely:

- The support of file formats by mobile platform iOS.
- The speed of processing and opening file formats for the iOS mobile platform.
- A set of supported file format functions for displaying three-dimensional models.

2.1 The support of file formats by iOS mobile platform

Collada and USDZ formats are supported by the iOS mobile platform at the system level. This means that for their reflection in the mobile application, developer does not need to use third-party libraries or other tools and respectively this simplifies the application's development process.

In the case with the Obj format, iOS allows developer to read information about the structure of models in this format using built-in tools. However, this does not apply to additional .mtl files containing information about the colors and textures of models. This, in turn, requires developers to create additional logic for the appropriate mapping of models or the use of third-party solutions.

As far as the Fbx format is concerned, this file format is not supported at the iOS OS level. In turn, Autodesk offers to use its own library for integrating the format into the iOS mobile platform - FbxSdk. However, this library is not compatible with the development environment for mobile applications for the iOS platform - XCode starting with version 10.0 (as of November 2019, the latest version is 11.2.1) [16]. A possible solution for porting a file to the iOS platform is the popular third-party library AssimpKit [17]. However, during the research of displaying models using this library, sometimes collisions or even premature termination of the program occurred, which may negatively facilitate on the appropriateness of using this format when developing a mobile application. So, there are ways to display files of this format for the iOS platform despite the lack of system support, but they are quite risky and should be compensated for by other advantages of this format for a specific task.

Therefore, from the point of view of convenience, working with Collada formats and USDZ files will not require the installation of additional libraries, which in turn saves the size of the final mobile application. If you use a third-party library AssimpKit, we need to consider the fact, it was developed by a group of non-profit enthusiasts who do not take responsibility for the reliability of this library.

2.2 Opening speed of file formats by the iOS mobile platform

Since each format of three-dimensional files structures information about a three-dimensional model in different ways, the processing time for the same model will be different, depending on the selected format. Because of this, a comparison of speed was performed of the same models for selected formats.

Models for comparison were downloaded from https://free3d.com [18]. Since among the compared formats, the Obj format does not support animation, two types of comparison were carried out:

- Comparison of the speed of opening models without animation of all formats.
- Comparison of the speed of opening models with animation among the formats Fbx, Collada, USDZ.

The comparison was carried out on two iOS devices, iPhone July 2016 release and iPad Pro 11 "2018 release, with iOS operating systems version 13.1.2 (the latest version as of September 2019).

To compare static objects, we selected a three-dimensional model of a military helicopter (46538 polygonal elements) in the Obj format, which, using the FbxConverterUI, Blender, and USDZToolset software, was converted to other comparable formats. The following Table 1 gives the comparison results.

Table 1. Results of the opening speed of a static model in various formats

Format	File Size (MB)	Opening Speed (sec) iPhone 7	Opening Speed (sec) iPad Pro 11'
Obj (natively)	4,5	0,39	0,25
Obj (AssimpKit)	4,5	1,05	0,86
Fbx	1,7	0,87	0,59
Collada	6,3	0,10	0,05
USDZ	5,0	0,53	0,40
USDC	2,6	0,47	0,27

To compare the model with the animation, a model of a warrior with a sword (3438 polygonal elements) of the Collada format was selected, after which, using the tools mentioned above, the input model was converted to other compared formats. The following Table 2 gives the comparison results.

According to the obtained results, the Collada format is the fastest to open. This is primarily due to system-wide format support. However, you should pay attention to the fact that models in this format take up more memory. This is due to the method of structuring and storing information, not in binary form, which are in the case of other compared formats, but in the XML markup language[20, 21].

When opening static files, the Obj format showed the second-best result. However, to display complex models with different textures, during testing, the third-party AssimpKit library was used. The experiments showed that in this case, the time costs significantly increase and exceed all other compared formats, which makes such a combination the least effective among all formats.

Table 2. Results of the speed of opening a model with animation in various formats

Format	File Size (MB/KB)	Opening Speed (sec) iPhone 7	Opening Speed (sec) iPad Pro 11'
Fbx	677 KB	0,30	0,24
Collada	1,2 MB	0,03	0,02
USDZ	866 KB	0,37	0,26
USDC	327 KB	0,10	0,20

USDZ files are supported at the system level in the same way as Collada. Binary and archived versions of USD files: USDC and USDZ, respectively, showed average results in the research. The time difference between these types of USD files is due to the fact that before processing USDZ, the format is previously unpacked. The archive also contains all the textures, increases the file size, but also adds convenience when working with the file, because all assets are stored in one place and it is easier to manipulate. Among the compared formats, USDZ is the only one that preserves assets in this way.

Fbx files showed the worst temporary results. In addition, Fbx files were the worst to convert. Also, during the comparison of various models, Fbx files often contained artifacts or even were not displayed correctly at all, despite the fact that the conversion took place using official software from Autodesk FbxConverterUI. All these factors make Fbx files an outsider among the compared formats.

Thus, according to the received results of the opening speed of the compared three-dimensional file formats, the Collada format is the undisputed leader, however, the USDZ format should be considered through the approximate speed and the lower file size, which will be useful when working with complex multi-polygon models.

2.3 Set of supported function formats

Although all compared formats support similar technology stacks, there are still differences between them. They can significantly affect the appropriateness of using one format or another when solving a specific problem. This section compares technology supported by formats. Among them:

- Description of model geometry using approximate and persistence meshes [22]. These technologies allow tessellation as well as precise coding of the geometry of curved surfaces that did not deform with a change in expansion. The USDZ format is also by default compatible with OpenSubdiv [23]. This is a set of libraries that allows dynamic speed and smoothing surfaces.
- Colors displaying. Usually they are represented by indicating the values of the red, green and blue components ranging from 0 to 1. Collada, Fbx and USD also support Alpha Blending, which allows you to specify the fourth parameter for the color - the alpha component. Possible values for it are from 0 to 1, where 0 is completely transparent and 1 is completely opaque.
- Textures displaying. All compared formats support texture mapping. Collada, Fbx and USDZ also support multitexture [24], which allows the simultaneous use of two or more texture as a template for surface models.
- Information about the scene. The Obj format does not contain scene information at all. Other compared formats support storing information about the camera light and relative position [25].
- Models animating. Collada, Fbx and USDZ support keyframe animation. Their implementation uses a sampler that describes how to manipulate the mesh using splines depending on the time elapsed. In addition, these specifications contain information on how to scan using the algorithm, and how to use the information stored in the respective nodes. According to Obj - this format does not support animation.
- File structure. File structure of the Fbx, Obj and USDZ formats can be stored both in ASCII and in binary form. According to Collada - it uses the XML markup language to store model information. In the case of the Obj format, the ASCII version (.obj) is independent and the binary (.mod) is proprietary.
- Documentation formats. Collada and UZD have a full official specification with usage examples. Fbx only contains class documentation and methods for the Autodesk Fbx-Sdk library. Obj does not have any official specification, however, due to the relative age of the format, there is a large number of independent, open sources.

The following Table 3 gives all the supported technologies of the compared formats.

Table 3. Supported technologies of the compared formats

	Obj	Fbx	Collada	USDZ
Approximate and persistence meshes	Supporting	Supporting	Supporting	Supporting
Colors and textures	Supporting	Supporting	Supporting	Supporting
Multitexture	Not supporting	Supporting	Supporting	Supporting
Scene information	Not supporting	Supporting	Supporting	Supporting
Animations	Not supporting	Supporting	Supporting	Supporting
File structure	ASCI/Binary	ASCI/Binary	XML	ASCI/Binary
Specification	Third party	Absent	Official	Official

So, considering the technologies supported by the formats, the most universal are Collada and USDZ due to good documentation and openness. As for the Fbx format, its main advantage among other formats is the large number of supported software from Autodesk.

Conclusions

This article analyzes the common formats for displaying three-dimensional models on the iOS platform, identifies their features advantages and disadvantages. In the course of work on this article, a mobile application for the iOS platform was developed to test methods for integrating file formats for building three-dimensional objects, namely Obj, Fbx, Collada and USDZ.

After conducting a detailed analysis of each format, analyzing the spectrum of their technologies, the support of iOS mobile platform, as well as practically exploring how to integrate them on the iOS mobile platform, the following final conclusions were drawn:

Obj format that supports the least technology stack. It is advisable to use it in projects where support for the scene or animation is required, and there is a need for support and consistent interpretation by various CAD software. It is supported by a large number of modern software for three-dimensional modeling. Simple models of this format are quite easy to display on the iOS mobile platform without using third-party libraries.

Fbx is a common solution for working with three-dimensional files. This is primarily due to the availability of a large number of software for creating models in this very format. However, files of this format, are possible but rather problematic to display on the iOS mobile platform, due to the lack of support for the format at the system level. Third-party solutions for integrating the format into projects for iOS, such as AssimpKit, do not always allow you to reflect the models properly.

The format should be used in projects where the costs of converting to another format exceed the costs of setting up the integration of the format and its proper reflection in projects for iOS. For example, when there are a large number of complex three-dimensional models that need to be integrated into the project.

Collada is a large and complex format that supports many features. Although the specification is a big plus when working with the format, the drawback is the lack of common standards for creating models, which implies a different approach to data structuring and possible conflicts / artifacts when displaying them. The format began to lose its relevance with the advent and development of a more standardized Fbx file, which makes its use in projects not promising and difficult to support in the future.

USDZ is the most convenient and relevant format for integration under the iOS platform. It supports a wide range of technologies, as well as:

- Contains built-in compatibility for smoothing surfaces using OpenSubdiv libraries, which allows to optimize the size of models without losing the quality of their reflections.
- Contains official documentation with examples of building elements.

- The only one of the compared formats that allows several designers to work on the same three-dimensional project without collisions in the files.
- Optimizes file size using the built-in link system and virtual copies of primitives.

During testing, this format proved to be better for use in projects under the iOS platform, both for displaying small static models and for manipulating several complex and animated objects. It is easily ported to the iOS mobile OS, and the official USDZ Toolset allows you to efficiently convert third-party formats that are not supported by iOS to USDZ format.

References

1. Obj File Documentation Homepage, https://www.fileformat.info/format/wavefrontobj/egff.htm, last accessed 2019/09/15.
2. Fbx File Documentation Homepage, https://www.autodesk.com/products/Fbx/overview, last accessed 2019/09/17.
3. Collada File Documentation Homepage, https://www.khronos.org/collada/, last accessed 2019/09/18.
4. USDZ File Format Explanation Topic, https://medium.com/techinpieces/all-about-apples-new-usdz-file-format-simplified-12dff29f3fc0, last accessed 2019/09/21.
5. Paiva P. V. F., Machado L. S., de Oliveira J. C.: Simulation and performance evaluation of a distributed haptic virtual environment supported by the cybermed framework. Virtual Reality (SVR), XIII Symposium on IEEE, pp. 188 - 196. IEEE, Uberlandia, Basil (2011).
6. Velev D., Zlateva P., Steshina L., Petukhov I.: Challenges of using drones and virtual/augmented reality for disaster risk managemen. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-3/W8, pp. 437-440. Gi4DM 2019– GeoInformation for Disaster Management, 3–6 September, Prague, Czech Republic (2019).
7. Dávidekováa M., Mjartana M., Greguša M.: Utilization of Virtual Reality in Education of Employees in Slovakia. Procedia Computer Science 113, 253–260 (2017).
8. Schuemie, Martijn J., Peter van de Straaten, Merel Krijn, Charles A.P.G. van der Mast: Research on Presence in Virtual Reality: A Survey. Cyberpsychology and behavior: the impact of the Internet, multimedia and virtual reality on behavior and society 4, 183-201 (2001).
9. Park Y., Rue S.: Analysis on Smart City service technology with IoT. Korea institute of information Technology Review 13(2), 31–37 (2015).
10. Kazarian A., Teslyuk V., Tsmots I., Tykhan M.: Implementation of the Face Recognition Module for the “Smart” Home Using Remote Server. Advances in Intelligent Systems and Computing III. AISC 871, Ukraine, Lviv (2019).
11. Park Y., Rue S.: Analysis on Smart City service technology with IoT. Korea institute of information Technology Review 13(2), 31–37 (2015).
12. Boreiko O. Y., Teslyuk V. M: Developing a controller for registering passenger flow of public transport for the "smart" city system. Eastern-European Journal of Enterprise Technologies Vol. 6, 3(84), 40–46 (2016).
13. Berezsky O., Verbovyy S., Pitsun O.: Hybrid Intelligent information techology for biomedical image processing. Proceedings of the IEEE International Conference of Computer Science and Information Technologies, CSIT’2018, 420–423 (2018).
14. Berezsky, O., Pitsun, O., Verbovyy, S., Datsko, T., Bodnar, A.: Computer diagnostic tools based on biomedical image analysis. 14th International Conference The Experience of De-

signing and Application of CAD Systems in Microelectronics, CADSM 2017, pp. 388 – 391 (2017).

15. Poniszewska-Maranda A., Kaczmarek D., Kryvinska N.: Studying usability of AI in the IoT systems/paradigm through embedding NN techniques into mobile smart service system. Computing, 1–25 (2018).
16. XCode Program Environment Homepage With Documentation, https://developer.apple.com/xcode/, last accessed 2019/10/11.
17. AssimpKit Open Source Framework Documentation Homepage, https://assimpkit.readthedocs.io/en/latest/, last accessed 2019/15/11.
18. Three-Dimensional Examples Webpage, https://free3d.com, last accessed 2019/10/11.
19. USD File Official Documentation Homepage, https://graphics.pixar.com/usd/docs/index.html, last accessed 2019/09/21.
20. Denysyuk P.: Usage of XML for fluidic MEMS database design. Proceeding of the 3rd International Conference of Young Scientists "Perspective Technologies and Methods in MEMS Design", MEMSTECH 2007, 148 (2007).
21. Denysyuk P., Teslyuk V., Khimich I.,Farmaga I.: XML application for microfluidic devices description. Proc of the IX-th Intern. Conf. on “The Experience of Designing and Application of CAD Systems in Microelectronics”, CADSM’2007, 567–569 (2007).
22. Precise Mesh Technology Explanation, https://www.spatial.com/products/3d-precise-mesh, last accessed 2019/09/20.
23. OpenSubdiv Technology Homepage, http://graphics.pixar.com/opensubdiv/docs/intro.html, last accessed 2019/11/11.
24. Multitexture Technology Explanation, http://what-when-how.com/opengl-programming-guide/multitexturing-texture-mapping-opengl-programming/, last accessed 2019/09/25.
25. Ivanov, Y., et. al.: Adaptive moving object segmentation algorithms in cluttered environments. In: Proceedings of 13th International Conference: The Experience of Designing and Application of CAD Systems in Microelectronics, 97-99 (2015)