A modern approach to storing of 3D geometry of objects in machine engineering industry

E A Sokolova, G A Aslanov and A A Sokolov

North Caucasus Mining and Metallurgic Institute (State Technical University), 44, Nikolaev str., Vladikavkaz, 362021, Russia

E-mail: katya_sea@mail.ru, gevork.ag@yandex.ru, asklv@mail.ru

Abstract. 3D graphics is a kind of computer graphics which has absorbed a lot from the vector and raster computer graphics. It is used in interior design projects, architectural projects, advertising, while creating educational computer programs, movies, visual images of parts and products in engineering, etc. 3D computer graphics allows one to create 3D scenes along with simulation of light conditions and setting up standpoints.

1. Introduction

To study the methods and means of composition, such as reproduction of a space, an environment, light-and-shade, laws of linear, air and color perspectives, the advantages of this kind of computer graphics over vector and raster graphics are quite obvious in this case. In 3D graphics, images (or characters) are modeled and moved around the virtual space, in a natural environment or in an interior, and their animation allows one to see an object from any standpoint, to move it in an artificially created environment and space, accompanying it with special effects [1-5].

3D computer graphics, as well as vector graphics, is object-oriented, which allows changing not only all the elements of a 3D scene, but also each object individually. This kind of computer graphics has great potential when supporting technical drawing. By means of graphic editors of 3D computer graphics, such as Autodesk 3D Studio, we can perform visualization of engineering details and products, and perform modelling of buildings and architectural constructions studied in the relevant section of architectural drafting. Besides, these sections of descriptive geometry as prospect, perspective and orthogonal views can receive graphic support as the principles of building 3D images are partly borrowed from them.

2. Materials and methods

For arts and crafts, 3D computer graphics enables prototyping of future products reproducing the structure and the texture of the materials of which these products will be made. The possibility to see the model of the product from any point of view before its production in the material allows one to change and correct its form or proportions, which can be no longer possible once the work was started (e.g., jewelry, ironwork, etc.). In the same way, 3D graphics can be used while creating sculptures, design, artistic graphics, etc. 3D animation and special effects are also created by means of 3D graphics. Creating video tutorials for training programs can be the main way to use these features of 3D computer graphics [1].

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Published under licence by IOP Publishing Ltd
Today, there are a lot of various file formats allowing one to store 3D geometry of objects. Let us analyze the three main types, their advantages and disadvantages [7-12,15]. One of the most popular formats is .OBJ. The .OBJ format is a match-all format for storing 3D objects. Files in the .OBJ format can be opened in all modern 3D editors and viewers, such as NewTek LightWave 3D, Adobe Photoshop CC 2015, Autodesk Maya 2016, DAZ 3D DAZ Studio 4, IMSI TurboCAD Pro 2016, Blender, MeshLab, MAXON Cinema 4D, Smith Micro Poser 11, etc.

3. The study of the structure of the files which contain 3D object geometry

Usually, the OBJ file format is a simple data-format that represents 3D geometry alone - namely, the position of each vertex, the UV position of each texture coordinate vertex, vertex normals, and the faces that make each polygon defined as a list of vertices [13,14]. This format is very convenient due to its simplicity, e.g. it is easy to read, and due to its transparent structure, it is easy to understand which geometry elements are described.

Basic elements of an .OBJ file:
- # is a comment (like // in C#)
- usemtl and mtlib describe the material library of the model.
- v is a vertex.
- vt stands for a vertex texture coordinate.
- vn is a vertex normal.
- f is a face (surface).

Another popular format is .3DS. A .3DS file consists of chunks, each of which contains some useful data and, possibly, sub Chunks. Most chunks contain either data or sub-chunks, although there are complex chunks. The general format of each chunk looks as follows:
Table 1. The total format of each unit

| Offset | Length | Data                              |
|--------|--------|-----------------------------------|
| 0      | 2      | Chunk id, chunk_id                |
| 2      | 3      | Chunk length, chunk_len           |
| 4      |        | chunk_len data or sub-chunks (depending on chunk_id) |

The list of the main .3DS-chunks:

- CHUNK_MAIN = 0x4D4D; // [-] main chunk
- CHUNK_OBJMESH = 0x3D3D; // [-] various objects
- CHUNK_OBJBLOCK = 0x4000; // [+ ] object block
- CHUNK_TRIMESH = 0x4100; // [-] trimesh-objects
- CHUNK_VERTLIST = 0x4110; // [+ ] vertices list
- CHUNK_FACELIST = 0x4120; // [+ ] faces description
- CHUNK_FACEMAT = 0x4130; // [+ ] face materials
- CHUNK_MAPLIST = 0x4140; // [+ ] mapping texture coordinates
- CHUNK_TRMATRIX = 0x4160; // [+ ] translation matrix
- CHUNK_CAMERA = 0x4700; // [+ ] camera
- CHUNK_MATERIAL = 0xAFFF; // [- ] material
- CHUNK_MATNAME = 0xA000; // [+ ] material name
- CHUNK_TEXTURE = 0xA200; // [- ] material texture
- CHUNK_MAPFILE = 0xA300; // [+ ] mapping texture file name
- CHUNK_KEYFRAMER = 0xB000; // [- ] animation information
- CHUNK_TRACKINFO = 0xB002; // [- ] object’s performance
- CHUNK_TRACKOBJNAME = 0xB010; // [+ ] object’s name
- CHUNK_TRACKPivot = 0xB013; // [+ ] object’s pivot point
- CHUNK_TRACKPOS = 0xB020; // [+ ] object’s track
- CHUNK_TRACKROTATE = 0xB021; // [+ ] object’s rotation track
- CHUNK_TRACKCAMERA = 0xB003; // [- ] camera track
- CHUNK_TRACKFOV = 0xB023; // [+ ] FOV camera track
- CHUNK_TRACKCamroll = 0xB024; // [+ ] roll camera track
- CHUNK_TRACKCAMTGT = 0xB004; // [- ] camera target track

The file represents a binary set of data, and in order to read it, one must be sawy to work with such kinds of files [3].

The last format that we are going to discuss is the .ASE format. Like .OBJ file, .ASE files are also presented as a text file, which facilitates the work with this kind of format.

- *MATERIAL_COUNT – number of textures used in a given 3D-scene.
- *MATERIAL – start of a material description block. It is followed by the number of the material.
- *BITMAP – the given material texture address.
- *MATERIAL_REF – reference to the number of the material assigned for the given object.
- Now let us study the description of geometry and texture coordinates.
- *GEOBJECT – refers to a new geometrical sub-object in the 3D scene loaded from the file.
- *MESH_NUMVERTEX – refers to the number of vertices that describe the geometry of the given 3D object (sub-object).
- *MESH_NUMFACES – refers to the number of polygons in the given geometrical sub-object.
• *MESH_NUMTVERTEX, *MESH_NUMTVFACES – describes texture coordinates and their bound to polygons.
• *MESH_VERTEX, *MESH_FACE, *MESH_TVERT, *MESH_TFACE – straight description of vertices and polygons [4].

4. Results and discussion
In the present study, to solve the given problem of developing a method of compression of digital scans lossless, a model of representation of spatial objects using an ordered one-dimensional representation of the point cloud of a 3D object is developed, where S – the point cloud, G – the curve that specifies the order of the points, R(G) – a scan function corresponding to the order of bypass of points G. The following functional description is a model-based representation of spatial objects using an ordered one-dimensional view.

Let us introduce the following notation:
O – 3D object,
b – bit pixel code,
M – memory in bytes
T – code table,
Tb(i) – the position of the code bits
B – block consisting of 3-bit items merger
x,y,z – a 3-bit header of the code in the block
Ts – table of offsets
i,j,k – codes, additional bits, the distance in the offset table.

For each three-dimensional object, we must specify the set of nodes and their connections with each other.

Each pixel is represented by a bit code. Then each of the codes must be placed in byte memory, while fulfilling the condition when each subsequent bit code is greater than the previous one.

The position of a bit code is being constantly checked when entering it to the storage.

Then, the resulting sequence is entered to the block, and, simultaneously with it, the information is analysed by constructing a bit tree, and on the basis of it — of the table of offsets. All the above-mentioned can be combined into a mathematical model:

\[
\forall O: \\
T \in \mathbb{M}(b)T, \text{ where} \\
\forall i \in (1,m): Tb_{i-1} < Tb_i \\
B = b_x \cup b_y \cup b_z \\
\forall Ts(i,j,k) = Tb \\
I(k) = \sum_{j=1}^{n} q_j(k) \\
x_0 = \min x, y_0 = \min y, z_0 = \min z
\]

(1)

Quantitative characteristics of products:
– the mathematical model which will allow processing images of 3D formats (*.obj*, *.3ds, *.ase )
– the operation time of the mathematical model depends on the quantity of processed images (directly proportional to the size of the uploaded file)
– the created method of lossless compression for representation of objet d'art obtained in the 3D format will allow us to get images without artefacts with a probability of 0.87.

A number of experiments were conducted, on the basis of which the graph of dependence of running time of the mathematical model on the size of the file was constructed (figure 2).

The appearance of artefacts depends on the number of recovered points on the object. Figure 3 depicts a histogram that shows the percentage of the recovered feature points and the probability of occurrence of artefacts. The proposed software allows viewing all 3 of the above mentioned formats. The developed software can be useful in the machine engineering industry. Due to the fact that the application is not overloaded with unnecessary functions, it can quickly display the necessary drawing
on all its sides. The digital model simulates the final product and gives engineers the ability to visualize, optimize and control the result before creating a physical prototype.

**Figure 2.** The graph of dependence of running time of the mathematical model on the size of the processed images.

**Figure 3.** Probability of artefacts occurrence

5. **Conclusion**

The application's main function is to work with the new format for storing 3D geometry, which is based on the basic principles of the .OBJ format.

The advantage of the new file formats is that they occupy much less storage space than .OBJ files do. According to our studies and experiments, the same object described by the new format file is about 28 times smaller than the initial one. This is due to the fact that this format uses a modern algorithm for lossless compression, which allows us to achieve this result. This feature will be very useful for the majority of users and machine engineers in particular.
Acknowledgments
This article was prepared by the results of the research carried out with the financial support of the Russian Ministry of Education in the framework of the basic part of the state task (project No. 3943 on the theme ‘Development of special mathematical software for the new structure of information analysis and the processing system for man-made cycles of enterprises of the mining and metallurgical complex’).

References
[1] Zhi-liang Zhu, Wei Zhang, Kwok-wo Wong and Hai Yu 2011 Inf. Sci. 6 1171-86
[2] Wei Zhang, Kwok-wo Wong, Hai Yu and Zhi-liang Zhu 2013 Commun. Nonlinear Sci. Numer. Simul. 3 584-600
[3] Xingyuan Wang, Lin Teng and Xue Qin 2012 Signal Process 4 1101-08
[4] Yicong Zhou, Weijia Cao, and Philip Chen 2014 Signal Process 8 197-207
[5] Kumaritov A M, Sokolova E A and Sokolov A A 2016 Mining J. 2 94-96
[6] Murillo-Escobar M A, Cruz-Hernández C, Abundiz-Pérez F 2015 López-Gutiérrez R M and Acosta Del Campo O R Signal Process 109 119-131
[7] Zhongyun Hua, Yicong Zhou, Chi-Man Pun C L and Chen Philip 2016 Inf. Sci. 10 80-94
[8] Yingqian Zhang and Xingyuan Wang 2014 Inf. Sci. 20 329-351
[9] Wei Zhang, Kwok-wo Wong, Hai Yu and Zhi-liang Zhu 2013 Commun. Nonlinear Sci. Numer. Simul. 8 2066-80
[10] Leo Yu Zhang, Xiaobo Hu, Yuansheng Liu, Kwok-Wo Wong and Jie Gan 2014 Commun. Nonlinear Sci. Numer. Simul. 10 3653-59
[11] Hongjun Liu and Xingyuan Wang 2011 Opt.Commun. 16-17 3895-3903
[12] Shannon C E, J. 28 1949 Bell Syst. Tech. 656-715
[13] Chen L, Jingyu Liu, Jisen Wen, Xiong Gao, Haidan Mao, Xiaoyan Shi and Qingling Qu 2015 Opt.Laser Technol. 6 80-86
[14] Yicong Zhou, Long Bao and Philip Chen C L 2014 Signal Process 97 172-182
[15] Ping Ping, Feng Xu and Zhi-jian Wang 2014 Signal Process 105 419-429