Mathematica
Balkanica

New Series Vol. 24, 2010, Fasc.3-4

File Format for Storage of Multimedia Information

Peter Armyanov

This article studies problems referring to storage, editing and streaming vector graphics and animation. The advantages and disadvantages of existing formats are reviewed and some suggestions to solve the problems are given. An approach for generating vector stream in a format, that allows embedding of additional media such audio and descriptive meta data is introduced. In parallel the problem of conversion multimedia files from one container format to another is described and solution for virtual conversion, based on file system filtering is mentioned. The ways for future development are pointed and some improvements of work already done are suggested.

AMS Subj. Classification: 68U05, 68P30

Key Words: vector graphics, stream transfer, computer animation, raster graphics, format-containers

1. Introduction

In the recent years vector graphics is gaining more and more popularity. It is in the basis of most of the engineering systems, geographic information systems and in the grounds of 3D modeling systems and has also well gained application in the profesional sphere. Vector graphic images are composed of multiple geometric elements, primarily points, lines and curves, analytically described. The basic advantages compared to the raster graphics is the ability for accurate and easy application of various transformations on the images, such as scaling or rotation [3]. One of the most well-known applications of the vector graphics is representation of 3D objects. With standard geometric transformations of a 3D object, described as a set of primitive elements, it may be represented as a two-dimensional image. In addition to that, without changing the description of the object itself, one may change the way a 3D object is constructed in two-dimensions by changing projection center or the projection plane for example. By applying various geometric transformations upon some components of the vector image, motion may be represented and 3D animation created. Representing the 3D animation as a description of a scene, composed of multiple independent objects and describing the motion with multiple geometric transformations, applied to a given subset of geometric primitives composing the scene, is used frequently when creating and editing a clip, but is inconvenient when reproducing on a computer screen, as it is a raster device [1]. Given this facts, to make representation easier usually the final clips are saved in a raster format, as a sequence of continuous images (frames). An inconvenience in this representation is, that the advantages of the vector graphics are lost for example, when zooming a given part of the image one cannot achieve the same accuracy and preciseness, as in vector representation. It turns out to be more convenient a vector clip to be saved in its original representation and before the representation of each frame to be rasterized. For a movement to be represented there are two approaches every frame to be saved as a vector scene or information about the transformations to be transmitted between two frames.

In the current moment there are few and undeveloped formats for representation of vector animation, especially when concerned streaming transmittance. The contemporary characteristics of the processors allow a fast and convenient processing of vector images the problems with this type of animation gain more attraction. This paper examines the problems evolving from the description of animation in a vector format and the various approaches for overcoming these problems [5].

2. Characteristics of vector graphics and animation

Every vector image is a set of geometric primitives points, lines, curves, often in more complex groups polygons, shapes, letters. For the description of these components their analytical mathematical description is used. This type of representation gives one of the main characteristics of vector graphics namely the means various affine transformations to be applied on the image, so that it does not loose its image quality. Such common transformation is vector scaling to practically enlarge the image infinitely with no quality loss as in raster graphics. The most well-known example are the symbols in computer systems for text editing. On the other hand uniting the primitive components lets a given group to be treated in a transformation as single and independent from the other primitives in the image. This enables easy description of effects simulating movement animation. The union of multiple primitive components enables easy changing the details of the image when being viewed to show or hide given details. Thus, it is possible the visualization to be adjusted according to the characteristics of the particular machine which performs the image processing [1, 3].

In succession with the advantages of vector graphics there are some drawbacks. The most significant of them is the complex method of representation on the screen. Because the screens are raster devices, in order to represent vector format graphics, it must be first transferred into a matrix of pixels. Taking into consideration the advantages of vector representation, as well as their gaining popularity, modern computer science gives special attention to the development of optimized algorithms for effective rasterization as well as developing special processors for this types of calculations. Another disadvantage is the comparatively huge amount of information, which has to be stored when describing complex and detailed images. While in raster graphics we have a rather constant amount of information needed to describe each image, when working with vector graphics it depends strongly on the type and detail quantity in the image. When transmitting an animation clip, represented in raster format, even with the usage of time compression and motion vectors, the size of each image of the clip, called frame, is relatively equal. In the vector approach for description, when the transformations between two frames are described analytically, the quantity of transmitted information decreases significantly. This is at the expence of the more complex process of representation of the clip. Thus when describing the clips in vector formats there are two basic approaches: describing each frame independently from the others, and describing the frame with multiple transformations, which differ from the previous frame. The advantages of the first approaches is the easier process of representation, as well as the opportunity to transfer the clip as a continuous stream. In the second approach the quantity of transferred information is much less, but the process of depicting is more complex first, the scene is generated, next the previous scene is made with the necessary transformations, after which its being rasterized on the screen. Another disadvantage is the impossibility the clip to be transmitted as a stream in order a scene to be represented, information is needed to be present for the current scene and the full list of all transformations frame by frame. This problem may be solved with the integration of the so-called key scenes, similar to the key frames in the MPEG raster compression. Thus, a stable balance between the amount of the transmitted information and the time, needed to collect information from the carrier stream can be achieved, before the clip itself is ready for representation [5].

There are various formats for representation and transfer of static images and video clips when it concerns raster graphics. When storing a video, two types of compression are mainly used. The first compresses each frame independently. In the other method part of the frames are compressed independently from the rest, for which only the part different from the previous or the following frame is described. The advantage in the first case is that the clips are easily modified they may be cut off at any given point. The disadvantage is the comparatively low quality of the image and the big size of data. In the second case it is possible to preserve the quality and decrease the amount of generated data. The disadvantage though is the inability to cut the clip at a random frame.

The multimedia data itself is merged and preserved in files containers. Up to this moment the multimedia containers can be divided into two basic categories depending on their purpose: formats facilitating the transfer and preservation of multimedia information and formats facilitating the editing (and mostly the assembly) of multimedia. Additionally, we can divide the formats into professional and user-oriented. The transferring formats may also be divided into stream and index.

The purpose of the streaming formats is to allow transmittance of continuous stream of information, so that users begin to read it at a random place without loss of multimedia. Because of this requirement, the describing information in this type of formats is repeated at a certain period in the stream. Such formats are mostly used in the digital television transmissions. They do not support a random access to the data. A typical representative is the MPEG format family for example mpeg2 TS which transmits satellite television. The indexed formats for preservation aim to permit an easy access to a relatively random time in the file. This is usually achieved by keeping the indexed table with records for each frame from the media. These formats are used most often as final storage formats and are convenient for representation or editing. Because the describing information is stored primarily at the beginning or at the end of the file, these files are difficult to represent unless a part of them is not accessed and thus cannot be used in stream transfer. Often these formats allow preservation of text media for example subtitles. Typical representatives are AVI, VOB, as well as the usage in the professional video representation format MXF.

The formats facilitating the editing are mostly created especially for a given professional system for video or audio assembly and thus contain multiple specific descriptive data. The most important between them is the information concerning the average time needed to record each frame, called timecode. These formats allow an easy editing and cutting the media in a random frame. They may also contain multiple different media of one type, for example several sections with video information in different compressions or formats. Many of the formats in this category may be considered as index transferring format. The most well-known formats of these types are MXF, Apple's MOV and its open version MP4.

A comparison of some well-known file formats based on the type of information they might contain and additional features is given in Table 1.

Container format	Supported video compression types	Supported audio compression types	VBR video support	VBR audio support	Inplace editing	User defined information	Subtitles	Streaming	Random access
3GP	MPEG-4, H.263, H.264	AMR, AAC	yes	yes	no	no	no	yes	no
ASF	no limitations	problems with Vorbis	yes	yes	no	no	yes	yes	limited
AVI	no limitations	problems with Vorbis	yes	yes	no	no	yes	no	limited
Flash Video	Sorenson, VP6	Raw, ADPOM, LPCM, MP3	no	no	no	no	no	yes	no
Matroska	no limitations	no limitations	yes	yes	no	yes	yes	yes	yes
MPEG-2 PS	MPEG-1, MPEG-2	MPEG-1, MPEG-2 AC3	yes	yes	no	no	yes	no	yes
MPEG-2 TS	MPEG-1, MPEG-2 H.264, MPEG-4	MPEG-1, MPEG-2 LPCM, AAC	yes	yes	no	no	yes	yes	no
MOV	no limitations	no limitations	yes	yes	yes	yes	yes	yes	yes
MP4	MPEG-1, MPEG-2 H.263, MPEG-4	MPEG-1, MPEG-2 Vorbis, AAC	yes	yes	yes	yes	yes	yes	yes
MXF	DV,raw,MPEG SD, MPEG HD,XdCAM,IMX	LPCM, AES3, MPEG-1, MPEG-2	no	yes	no	yes	no	no	yes
OGG	no limitations	no limitations	no	yes	no	yes	yes	yes	yes
VOB	MPEG-2 part 2	AC-3 LPCM MPEG-1, MPEG-2	yes	yes	no	no	yes	no	limited

TABLE 1. Comparison of some multimedia formats

Because of the different characteristics of the given formats it is acceptable for a given clip to lose part of its information when being transformed from one format to the other and this conversion to be impossible without trans coding this clip to other formats.

One possible solution to this problem is examining the creation of virtual multimedia files, representing the video or audio data in a format different from the one they have initially been saved in, without being represented in this virtual format in a physical medium. A copy may be placed on the media in one format, and the others to be derived while processing virtually with file system level filters. This strategy benefits from the fact that modern computers have powerful processors and huge operating memory. For speeding up the creation of the virtual files it is necessary some preliminarily processed information to be kept for the original file in order to make the creation of the virtual files and the work with them more efficient and quicker. In [6] a detailed information is given on the necessary preliminarily processed information for multimedia files. It also describes the storing format so that it can be easily created physical or virtual copies of a given multimedia file in various formats.

3. Basic formats for storing and transferring vector animation

In the current moment there are several formats for description of vector animation. The most frequently used one is the open format SVG (Scalable Vector Graphics). It serves as a basis for the more wide-spread amongst the users Macromedia's SWF (Small Web Format), later bought by Adobe. In this format the description of motion is implemented by affine transformation description. In this current form it does not allow the stream transfer of clips. Because of the huge user interest in the transmittance of the clips in Internet an expanded format for stream transmittance is developed - Flash Video (flv). This extension transmits the data in a raster form. SVG is a text format when data are transmitted as a XML document. This representation is chosen because of the simple formal description and interpretation, and the possibility for integration, as well as its portability in spite of the differences in the physical representation of the various hardware platforms. To decrease the size of the information transmitted the format allows compression of data. One disadvantage of this format is that it does not support the transition of non-graphic information, for example sound [3].

Another well-known format is Apple Animation. It describes the clips either as a set of transformations, or as a sequence of independent scenes. There is no approach for merging both descriptions. It supports RLE compression. The data are transmitted in a special binary representation. The format allows parallel transmittance of sound or raster described clips. A serious disadvantage is that its description is closed and it is bind with a series of formats, patents of Apple Inc, that is why it is not much popular in Europe. It does not support a stream except an annotated MOV container [2].

In the end of the 90's of the last century, the company Silicon Graphics started developing the standard for vector graphics OpenGL (Open Graphic Library), which in the moment is the most wide-spread interface for programming of vector graphic, preliminarily 3D models and CAD systems. Approximately at the same time GLS (Graphic Library Stream) was developed, in order to transmit the tracing commands in a stream, so that OpenGL could be used as a format for storing and transmittance of graphics and animations. It has both text and binary described version. A huge advantage is the fact that the commands are directed straight into the GPU (Graphical Processing Unit) of the video-card. Because of the little interest of programmers and users at the time OpenGL was invented and also the slightly-developed graphical processors at that time the development on this format were ceased soon after its creation [4].

Summarizing the characteristics of the well-known formats at the moment for representation and transmittance of vector graphics animation we notice the lack of stream formats for transmittance as well as the inefficiency and restrictiveness of the approaches, chosen for representation of information in comparison with the stream formats, used to transmit raster clips.

4. Approaches for effective description of streaming vector animation

The main fields in optimizing the description of the stream vector graphic are the size of the information transferred, the efficiency of the representation process, the opportunity for continuous stream transfer and the opportunity for transfer of additional multimedia information.

The simplest and yet most convenient approach for finding the size of information is by using compression. Modern processors have enough power, so that the decompression of the data stream does not reflect in slowing down the representation process. On the other hand, when using a compression algorithm without dependency usage it is possible to transfer data in a continuous stream. When using compression dictionaries it is possible that they are transmitted periodically in the stream together with the rest of the descriptive information [3].

Another approach is increasing the number of graphical primitives by using the so-called macros. These are means for parametric description of more complex graphic objects, mostly in three-dimensional graphics. Thus a cube may be represented as a set of 12 lines, describing its edges, each defined with its coordinates in both ends and as coordinates of a single vertex, orientation of a given edge and length of the edge. The same may be related to a set of more complex, but often used in 3D modeling and component animation. The increasing of the graphical primitives with such macros should not lead to less efficiency in representation, as well as not be affected by the transmission of data as a continuous stream [1].

A summary of the mentioned approach is the idea for introducing of specific for the given clip catalogs of the parametric objects a whole element from a scene may be described with such a catalog, parametrized at least by its position, orientation and size. When we want to include such an object in a given scene we simply quote the catalog record with a unique identifier (for example number) and we chose its parameters. This approach may decrease multiple times the size of the transmitted information, again with minimal delay of the visualization process. One disadvantage is the necessity of transmitting periodically these catalogs when making a continuous stream, which even when using compression may lead to a drastic rise in the size of transmitted information. Another, more serious disadvantage is the algorithmic complexity of the attachment of the primitives in a scene into an entity parametrized objects. When the scene is being drawn by a specific software, it stores data for these objects, but if the logic for the specific catalog is not compatible with the specific application, for example a codec processing a stream of vector graphic, finding and summarizing and parameterizing of objects pose serious algorithmic problems [1].

Another approach for decreasing the size of information, often used in computer animation is transformation macros. This approach is used when transmitting the animation as a set of continuous transformations which are applied to the objects in the main scene. Macros here are a brief parameterized description of a frequently used set of continuous elementary transformations, as well as repeating for example it is given that the last few transformations are repeated again and again a certain amount of time. This approach is particularly suitable for describing smooth movements in animation each frame differs from the previous by the position of an object moved with a fixed vector. Again, a summary may be made of this approach for more complex transformations, specific for the given clip, described in catalogs, but here the filtering and extracting of the same information for a record in the catalog after the clip is finished is practically impossible.

The application of some or all of the listed approaches when describing a vector animation leads to multiple decreasing of the size of information, describing a given clip by its main description with basic primitives without a serious loss of productivity in representation. This is a wanted effect, mostly when transferring the clips in a network with limited transfer speed and at the current moment several big companies have aimed their attention in this direction.

5. Characteristics of continuous video streams

With continuous video streams the client starts receiving information when attached to the stream at any random moment. Is it essential that when he starts collecting data after a given period of time to be in condition to begin reproduction of the clip in the stream. These streams are most often used in digital television, where, on the contrary, the information is transmitted only in raster representation. In the current moment continuous streams of vector graphics have broadest usage in development of rich Internet applications [5].

A basic requirement to the continuous streams is their descriptive information to be transmitted in a certain amount of time, especially chosen so that it does not affect the stream at one hand and on the other, so that the user will not wait too long for the information to arrive on the stream because without it, the reproduction is not possible. When this information is small in size it is acceptable to be repeated in short intervals of time(under 300 ms). If the information has a bigger size, it is divided into smaller parts with different priorities. Thus the information with highest priority, without which the representation is impossible, is repeated quite often, usually in every frame, the less priority information rarely and the user data those which have no direct relation with the clip representation are repeated in largest time intervals once in 3-4 seconds [5].

Often to the main carrier stream of video information other streams are added, carrying different media types sound, for example. It is possible these different streams to be carried in a parallel manner into different physical channels, or they might be mixed into one channel. If we use a parallel transfer, the descriptive information is usually placed in a separate stream, even if it is specific for each of the media streams. The parallel carriage allows the video clip to be transmitted at the same time in several streams, differing by the detail of the image and the user may choose which of them to represent depending on the characteristics of the carrier environment used or depending on the computer configuration. It is also accepted that dynamic change in the representation of the channel in a given moment may occur without its representation itself. Such a technology is realized in Apple's annotated multimedia streams [2, 5].

6. Summary

In the contemporary applications for vector graphics procession more and more features for creation of animated clips are invented. The presence of a small number of standard formats for description of vector animation imposes the clips to be stored in raster formats. The current paper examined the most-famous formats for storage of vector animation, as well as some multimedia containers. The characteristics of vector representation originate some specific problems concerning the developing format inefficiency in the size of needed data for storing or transmitting clips, as well as little or no options for streaming their representation. There are some well examined approaches for solving the basic problems, which arise when representing animation in vector format. The aim of the author is to develop a decision which gives streaming to vector graphics, which copes with the problems of the previously listed solutions.

References

[1] J. Vince , *Vector Analysis for Computer Graphics*, Springer, 2007

[2] *QuickTime File Format Specification* , **7** , Apple inc., 2009

[3] A. Adam , *SVG. Scalable Vector Graphics* , Franzis Verlag, 2002

[4] C. DunWoody, *The OpenGL Stream Codec* - Silicon Graphics - 1996

[5] B. Gilmer, *File Interchange Handbook for images, audio and metadata*, Focal Press, 2004

[6] P. Armyanov Minimal set of metadata for virtualization of multimedia files - In *Third International Conference on Information Systems and Grid Technologies, 28-29 V 2009, Sofia, Bulgaria,* 46-53

Sofia University St. Kliment Ohridski
Faculty of Mathematics and Invormatics
5, "James Bourchier" str,
1164 Sofia, Bulgaria

E-Mail:parmyanov@fmi.uni-sofia.bg