INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11
MPEG2004/M11374
October 2004, Palma de Mallorca, ES

Status: For consideration at the 70th MPEG meeting
Title: Implementation of LASeR and SVG in GPAC
Author: Cyril Concolato (ENST), Jean Le Feuvre

1. Introduction

This document presents part of the work[1] done to implement the LASeR specification within the open-source GPAC Project[2]. The GPAC project is an open-source project whose goal is to develop an open-source software suite to create and play-back MPEG-4 Content, with a particular emphasis on MPEG-4 Systems content. The current GPAC release (0.2.0) already covers a very large part of the standard, and features what can probably be seen as the most advanced and robust 2D MPEG-4 Player available worldwide, as well as a decent 3D MPEG-4/VRML player (CVS only). GPAC is currently running under Windows, Linux and WindowsCE/PocketPC platforms.

In the following section, this document describes the implementation status of the different modules implemented within the GPAC project. The third section describes the future work to be done and the last section gives some screenshots of SVG/LASeR content played in the Osmo4 player.

2. Status of the implementation

2.1. LASeR CD implementation

Our work in the course of the DANAE project and in the context of the LASeR CEs has consisted in adding to the GPAC project several modules in order to play SAF content with a LASeR scene description stream. The work has been focused to integrate the SAF/LASeR as much as possible into the GPAC architecture, that is to say to as much as possible in an MPEG-4 terminal architecture. Therefore the following modules were developped:

- a SAF plugin,

[1] This work has been partly financed by the european commission in the course of the IST project DANAE: http://danae.rd.francetelecom.com
[2] http://gpac.sourceforge.net, GPAC project

- a LASeR decoder,
- and a LASeR Renderer.

No encoder has been provided at the moment.

The SAF plugin is a file input plugin to an MPEG-4 player. It implements an interface close to the DMIF Application Interface (DAI). This plugin takes as input SAF stream and provides as outputs elementary streams including an Object Descriptor stream created on the fly based on the SAF Media Header. Moreover, at initialization, this plugin creates an MPEG-4 IOD with two ESD: one for the scene description stream (LASeR) and one for the virtual OD stream.

The LASeR decoder is a scene description stream decoder registered in the MPEG-4 terminal to decode streams of type Scene Description Stream and whose OTI is 0x09. This decoder takes as input LASeR acess units and a pointer to the scene graph. The result of the decoding is directly applied to the scene graph (LASeR commands). The current version of the LASeR decoder in GPAC is compliant with the LASeR CD issued in Redmond. It decodes all the test sequences provided with the reference software.

The LASeR Renderer is the module responsible of rendering the LASeR nodes (shapes, text, …) The current implementation is based on all the different modules of the GPAC project (Graphics Driver based on Microsoft GDI or on FreeType; Video and Audio Driver based on DirectX or SDL, Font Renderer based on FreeType, …). Among all the features of the LASeR CD all the primitive shapes have been implemented including text, the animation either using animate nodes or LASeR Commands have been implemented. The interaction, media handling and save/restore/clean features have not been implemented. Instead, support for compliant SVG has been added to GPAC.

2.2. SVG support in GPAC

Since the evolution of LASeR as currently specified in the study of CD implies a greater overlap between LASeR ans SVG, it has been decided to add the support for SVG into the GPAC project.
The current modules of the GPAC project which currently deal with SVG content are:

- An SVG textual parser,
- An SVG Renderer

The textual parser for SVG has been developped to add the capability to read SVG files and play them directly. The parser is based on LibXML[3]. It takes as input a conformant SVG file and directly modifies the scene graph. It is similar to the LASeR Decoder described in the previous section.

The SVG renderer, similarly to the LASeR render, allows the rendering of the SVG scene tree. Currently the following SVG elements are supported: rect, circle, ellipse, line, polyline, polygon, path (no elliptical arc) and text (partially), g and svg. All other nodes

[3] http://www.xmlsoft.org/, The XML C parser and toolkit of gnome

are ignored. Developpements are still ongoing. The following section describes the future work in the coming weeks and the last section shows some SVG snapshot to illustrate what can be played at the moment.

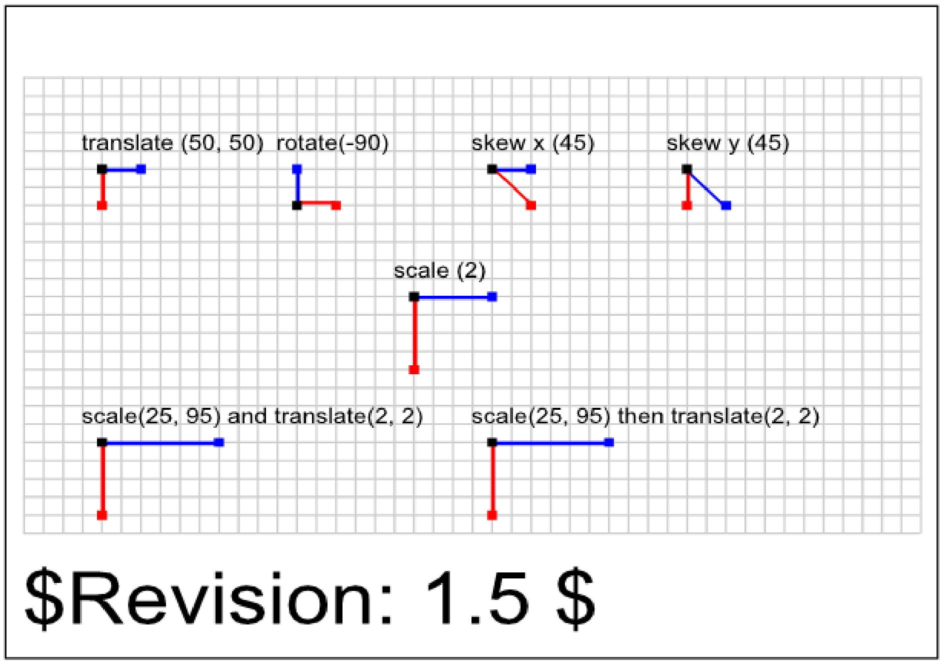
3. Future Work

The current working plan is to implement the following SVG nodes to cover as much as possible what can be done in LASeR and SVG:

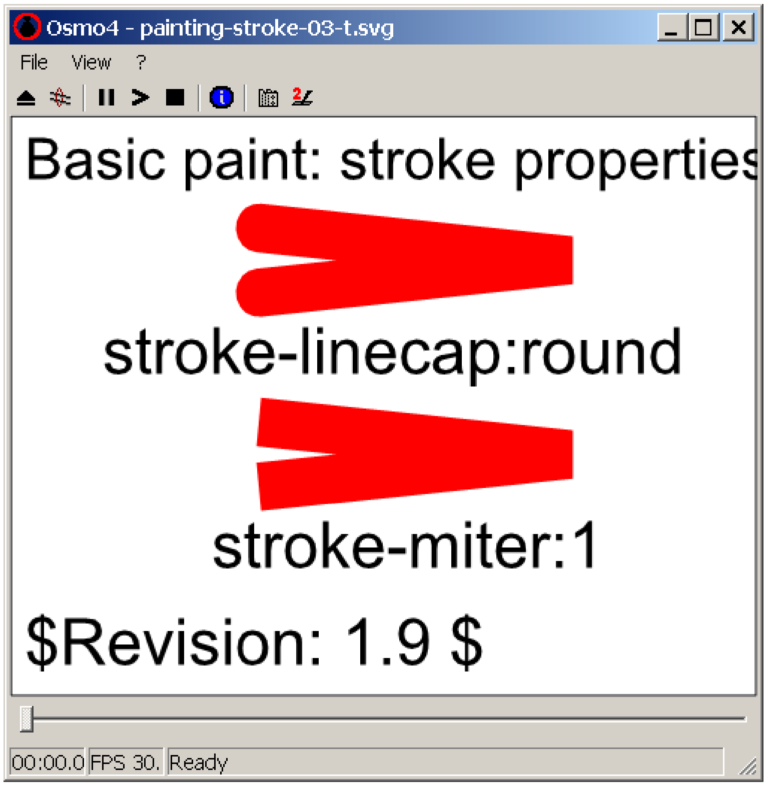
- animation elements: set, animate, animateTransform, animateColor, animateMotion: these elements though simple in their definition are quite heavy to implement but bring a lot of functionnality to the SVG content.
- video/audio: we intend to include support for media within an SVG. This should be done quite straightforwardly since GPAC allows the handling of several media types (MPEG-4 video/audio, MPEG-2, JPG, PNG …). This work should validate the fact that LASeR and SVG content work the same with media.
- Interactivity: the models are different between MPEG-4 Systems (BIFS, VRML) and LASeR/SVG in terms of interactivity. The challenge will be to unify the two approaches.
- If time permits, scripting will be added.

4. Screen Shots

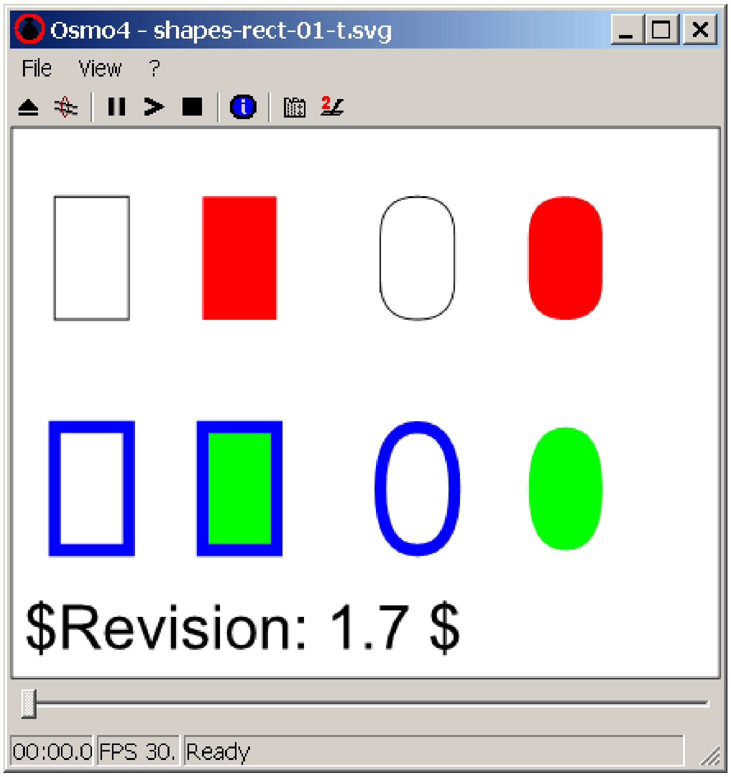
These screen shots represent SVG content played within the Osmo4 player, running on Windows XP. The sequences are taken from the SVG 1.1 test suite, without any modifications.



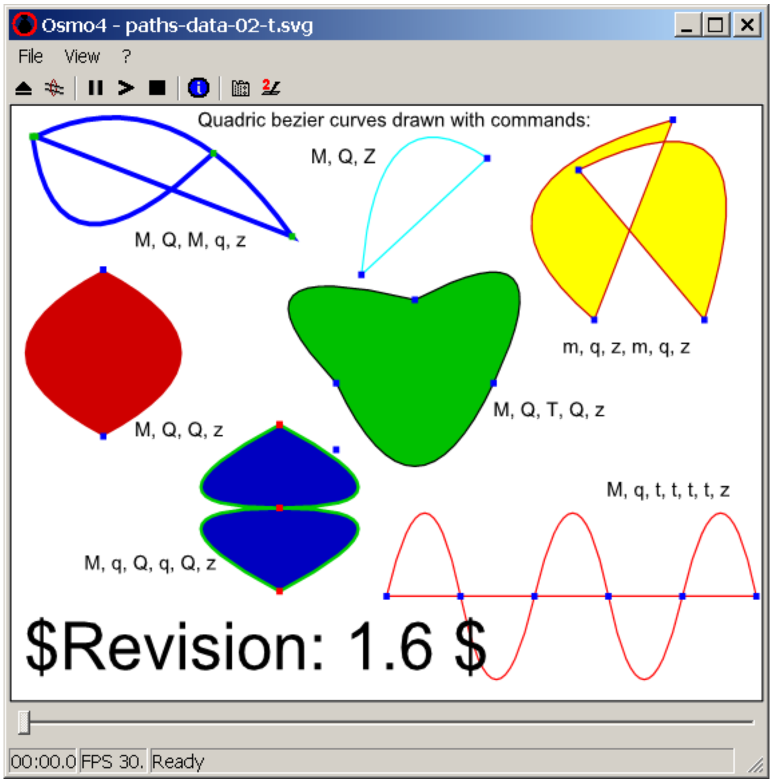
Coordinate transformations à la SVG



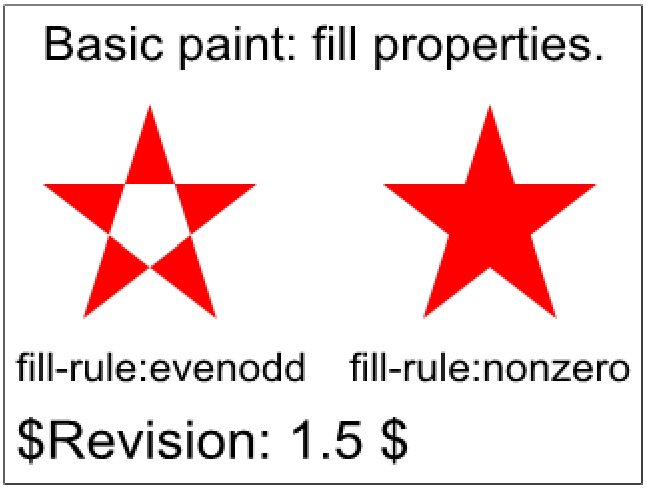
Paiting strokes à la SVG



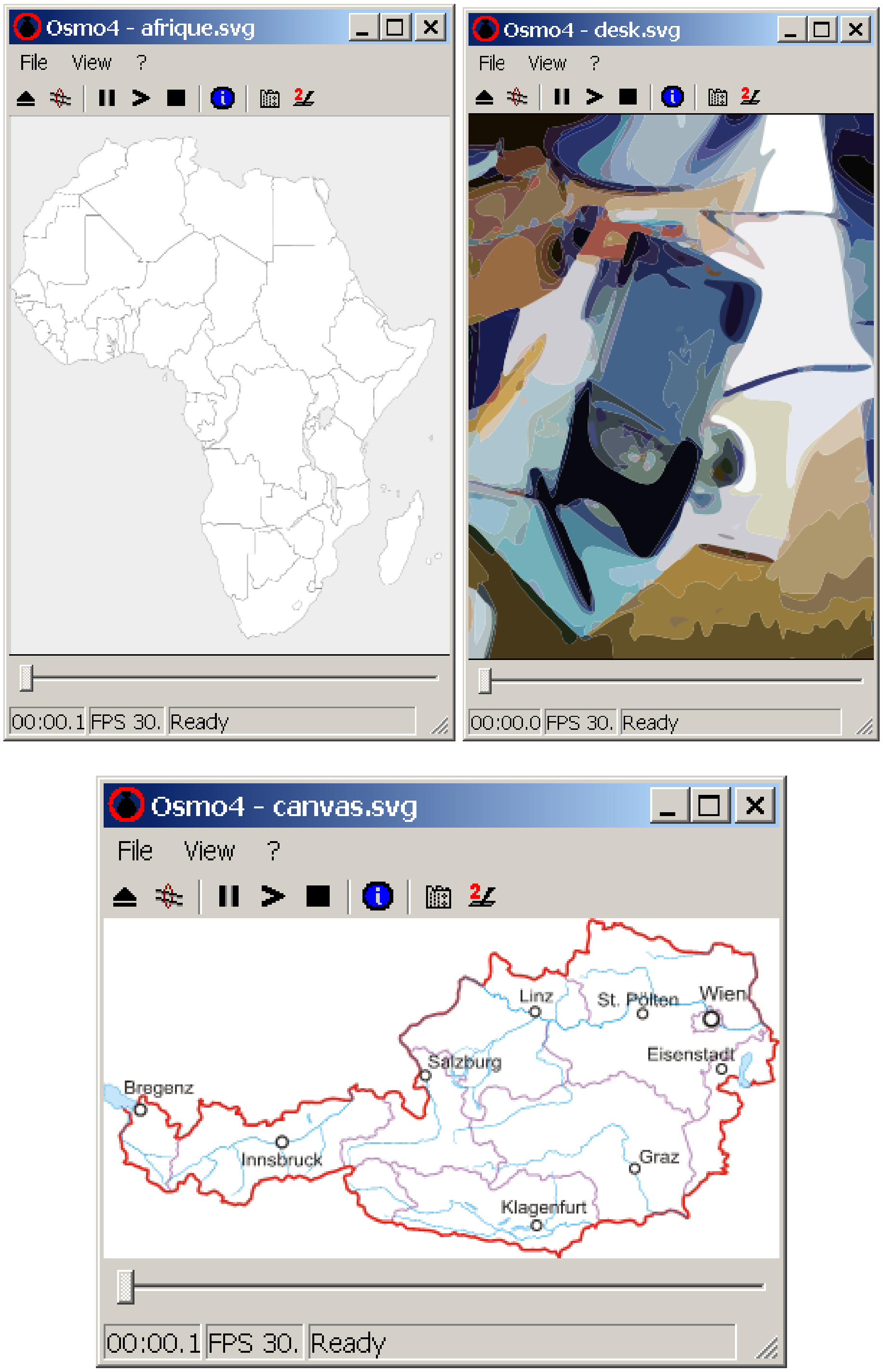
Rectangles including rounded ones.



SVG Paths, including bezier (quadratic and cubic)



Filling with fill-rule



Tests from the LASeR test suite