

MUAN: Animation for the rest of us

Margareth Catoia Varela Luiz Velho Hedlana Bezerra Bruno Madeira
Instituto Nacional de Matemática Pura e Aplicada - IMPA
Estrada Dona Castorina, 110, Jardim Botânico, Rio de Janeiro, RJ. 22460-320

Marcos Magalhães
Anima Mundi
Rua Elvira Machado, 5, Botafogo, Rio de Janeiro, RJ. 22280-060

Abstract

MUAN is a Stop Motion Animation System. The MUAN system allows animation movie production offering an integrated kit of hardware and software. It consists of a computer with Linux, a video camera and support over necessary accessories for interconnection. The software was developed by the team of VISGRAF Laboratory (IMPA), in cooperation with ANIMA MUNDI and IBM Brazil, and consists of programs to create, edit, manipulate and visualize animations.

1. Introduction

Although stop motion animation is an old technique, it was the base for evolution of the field and is being used nowadays due to its simplicity and intuitive principles: still images (stop) turn to movement (motion). Nowadays, modern techniques makes the computer an important tool applied in many areas. Thus it brings the possibility to build applications to create and manipulate animations with a camera easily attached, reducing the time and cost of production. Within this context, MUAN¹ is a free stop motion animation system that appears to be a simple alternative, easy to use, practical and cheap, supplying virtually all needs of an animator, professional or not, providing powerful tools to create, test, edit and play stop motion animations.

2. System Architecture

The MUAN system was developed for the Linux platform. The software part uses libraries or program codes that are freely distributed, making the application very flexible

for evaluations. The main idea of the system MUAN is to show images coming from a camera video stream, allowing the user to select the desired images. As the result, a set of captured/selected images will form an animation.

2.1. System Description

The MUAN software provides very important features of Stop Motion like toggling between stored and live frames, instantaneous preview of animation, delete/insert of frames, frame markers and frame flipping. This last feature is very useful for the Stop Motion animators because it allows to compare the last frame with the image coming from the camera (live incoming frame) so the user can check if the frame to be stored is the desired one and can adjust the scene accordingly. Frame markers are used to modify the duration (delay), to play or delete an interval of frames. By duration we means the time that the frame will be repeated within the animation sequence.

To provide Stop Motion features, we create an image list to store the frames of the animation, that is, the captured images with their properties like delay, time stamp, duration and position at the list. With a list, we can manipulate positions, visualizing any frame of animation at any time. Thus, the user interface provides components for playing or stopping animation, rewind, forward, go to first, to last or to an arbitrary frame position on the list, and clear all the list or remove some or several stored frames. Besides, the user can choose a camera mode, showing live frames, or memory mode, showing the captured images. In both modes, images are displayed in an OpenGL window, so the image to be shown is an array of pixels in RGB color. But in the list, the stored frames are saved in DV format. Although this format contains image and sound information, its length is more compact than a simple array of pixels.

Because the application has to show live frames coming from the camera, we create a thread which continuously get images from the camera in real time. When connected via

¹ Muan means firefly in tupi-guarani, a Brazilian indigenous language.

firewire, the application establishes a communication with the camera and grabs DV data. Images in DV format have 3:2 aspect ratio, that is, widescreen. For our purposes, we are not working with sound, we are focused only on capture images, so the captured DV frame has its sound track erased. When the connection with camera is over a composite or s-video input, application opens a direct communication with device, through video4linux interface, capturing images as an array of pixels in 24 bit RGB palette. In this case, captured images have 4:3 aspect ratio, so we put a black band on the laterals of the image and convert it into DV format to put it in the list of captured frames.

Once all the desired frames were captured, the user can save the animation asking the application to generate a video file. MUAN can record video files in AVI or MPEG formats. Another option is to save each frame as an image file in JPEG format. Thus, we also provide functions to read avi or mpg video files and ras or jpg image files. User can open one of those files and edit or can insert files of these formats in an animation that is being created.

Options like view image window in full screen, play the animation continuously or in loop and flip operation, that alternate camera mode and memory mode providing helpful features for the animator to analyze the last or next frame to be captured, are also present in user interface, making the application more friendly.

2.2. System Interface/Structure/Requirements

In order to provide the functionalities listed in section 2.1, we adapted some libraries commonly used on Linux platform that are freely distributed. We create a video library that implements functions of interface with I/O communication using the linux libraries/drivers according to the type of connection (firewire or v4l). In the case of firewire connection, the interface with iee1394 was based on IEEE1394Reader class which is part of Kino[5] application and asynchronously grabs DV data through the libraw1394. In the case of v4l connection, video library interacts directly with v4l driver. To implement the functions that store the animation in video files, we use the libavcodec and libavformat libraries, which compose the FFmpeg Project[1]. FFmpeg has functionalities to record, convert and play stream audio/video for most operating systems, including Windows. With these libraries, Muan is able to read and write MPEG files. To import JPEG images to an animation or export the animation frames to JPEG images, we use libjpeg[2] which is a widely used free library for handling the JPEG image format, written and distributed by IJG. The class ImgList implements the animation representation with a list used to store frames. To save the frame as DV data, like it comes from digital devices through a firewire connection, we implemented a representation based

on Frame Class from Kino, which provides utilities for processing digital video frames through the libdv. Libdv[3] is a software codec for DV video developed according to the official standards. Another widely used free C library that we use in MUAN is libsndfile[4]. Libsndfile is a simple and easy to use API for reading and writing a large number of file formats containing sampled sound and can be usable on Unix, Win32, MacOS and others. Although we are not working with sound for animation, sometimes we need sound on the user interface to provide a better and easier interaction.

3. Conclusions

The MUAN System was designed in conjunction with people who works, uses and does animation, that is, professional animators and educators. Due to this partnership, we have a robust system, tested by many users in real situations and by animators in their daily work. As successful experiences we can cite the Anima Mundi Festival and the Anima Escola project. At Anima Mundi Festival, MUAN is used in the open and free workshops that allow the spectator to become the creator, producing animations from clay, drawing, sand or cut-outs. Through Anima Escola project, animation workshops are offered to students and teachers of municipal schools of Rio de Janeiro, Brazil. Then MUAN integrates a technological system which allows the use of animation in schools, being a valuable instrument contributing to improve the quality of public education. Through a creative use of available technology, MUAN gives children an innovative and exciting learning experience.

The technological solution brought by MUAN makes the animation process very simple and fast, making it usable also by professional animators who want a quick result for a test or video-assisting tool in the production of stop-motion animation.

The open source version of MUAN software are already available. The installation files (binary distributions and source code) and additional information like documentation, manual, FAQ and installing instructions can be found at <http://www.muan.org.br>.

References

- [1] F. Bellard. Ffmpeg multimedia system, 2004. <http://ffmpeg.mplayerhq.hu/>.
- [2] IJG. Libjpeg, 2001. <http://www.ijg.org/>.
- [3] C. Krasic and E. Walthinsen. Libdv, 2003. <http://libdv.sourceforge.net/>.
- [4] E. Lopo. Libsndfile, 1999. <http://www.mega-nerd.com/libsndfile/>.
- [5] A. Schirmacher. Kino, 2002. <http://www.kinodv.org/>.