MUAN: A Stop Motion Animation System

Margareth Catoia Varela[1], **Luiz Velho**[1],
Bruno Madeira[1], **Hedlena Bezerra**[1], **Marcos Magalhães**[2]

[1]Instituto Nacional de Matemática Pura e Aplicada (IMPA)
Rio de Janeiro, RJ – Brazil

[2]Anima Mundi
22.280-060 – Rio de Janeiro, RJ – Brazil

{meg,lvelho,madeira,hedlena}@impa.br, marcos@animamundi.com.br

Abstract. *MUAN is a Stop Motion Animation System for Education. The MUAN system allows animation movie production offering an integrated kit of hardware and software. It consists of a computer with Linux, a digital video camera and support over necessary acessories for interconnection. The software was developed under Linux by the team of VISGRAF Laboratory, 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 and was the base for the ones that came after, it is being used nowadays due to its simplicity and intuitive principles. These days when the computer is an important tool applied in many areas it brings the possibility to build applications to create and manipulate animations with a camera easily attached, reducing the time and cost of their production. Within this context, MUAN[1] is a free stop motion animation system that appears to be a simple alternative, easy of use, pratical and cheap, supplying practically all needs of an animator, professional or not, providing powerful tools to create, test, edit and play stop motion animations.

1.1. Overview: the Development of Animation

Animation is the art of creating movement through a series of fixed images. Although along the History many different manifestations (such as paintings, sculptures, machines, toys and gadgets) already foresaw the possibilities for an illusion of movement, the first to achieve a universally convincing result was the Belgian physicist Joseph Plateau (1801-1883) who invented the phenakistoscope ("spindle viewer") in 1832. This simple machine used a sequences of 10 drawings disposed on a rotating circle with slots (an adaptation of the stroboscopic effects of the Michael Faraday's Wheel). With it, Plateau was able to prove his theory of the "persistence of vision", that until recently was considered the basic explanation for the whole cinematic phenomenon (including film and television). Plateau then stated that, if a sequence of fixed images, one slightly different from the other, are substituted in front of our eyes at a rate of 10 or more per second, we will perceive then as one single moving image. Besides the great curiosity and popularity provoked by the phenakistoscope. it has led to further inventions based on its concepts,

[1]Muan is the name of a firefly in tupi-guarani, the Brazilian ancient indian language.

such as George Horner's "Zoetrope" (1834), Emile Reynaud's "Praxinoscope" (1877) and the Lumière Brothers' "Cinematographe" (1895), this last one integrating the use of realistic photographic images, taken in a constant rate of more than 10 images per second.

So, even if the Cinema history has started with the use of drawn images in Plateau's first animated "cartoons", that facet was soon forgotten with the advent of "real" live action scenes being projected on a big screen. But, as soon as some artists discovered they could turn the crank of the camera just enough to impress one frame at a time, they realized they could do magical tricks with it, recreating or subverting the possible "real" movements of the objects in front of the camera. That's how animation was reborn in the cinema medium. One emblematic pioneer is American James Stuart Blackton, who did his first experiences using stop-motion. In "The Haunted Hotel" (1907) he was able to make dishes, knifes and forks perform as living actors in a hotel kitchen. After that, Blackton, who was a talented draftsman, soon discovered also that he could use the stop-motion technique applied to drawings instead of objects, and started producing some of the first movie cartoons in the history of animation. Due to its popularity (often linked to characters of the comic strips, another newborn art form), the drawn cartoons have become, along the 20th Century, the most popular technique for what is generally known as "animation".

1.2. Animation Applied to Education

The MUAN system was originally designed to fit the particular needs of a methodology of teaching animation as an educational language. This methodology has been developed by Anima Escola, the educational project put on by Anima Mundi, the International Animation Festival of Brazil. Anima Escola and Anima Mundi are detailed in Section 5.

The animation workshops of Anima Escola aim to make the animation language accessible and easy to use in the classroom. Intuitive and expressive materials and tools such as plasticine, paper cut-outs and body expressions are applied to stop-motion techniques to create animation scenes, in the fastest and easiest way possible. In this way, the student learns to use animation as a natural and familiar language, and not as an intricate and expensive technology as the common sense used to classify it.

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

1.3. Stop Motion Techniques

Stop Motion Animation is the same basic principle used to create a cartoon (drawn or cell animation). The very term that names it comes from the fact that every image composing an animation must be a still, fixed image. Still images (stop) turn to movement (motion).

An animation movie camera, therefore, must act more like a still camera that shoots, in sequence, one picture for each animation frame (be it 2D drawings or real 3D objects or living beings). The animator positions all objects or characters in a scene, takes a picture, moves an object a little bit and then takes another picture, and redo these steps until he finalizes the desired movement. These pictures are called frames of the animation movie and when we play all frames in order, we have the illusion that objects or characters are moving on their own.

2. Image Capture Technologies

Capture image is the first step to construct a Stop Motion Software. We can capture images from a camera or camcorder and transfer them to a PC using a video capture device for PCI bus, USB or IEEE-1394. Analog video capture devices connect cameras to computers using composite or s-video inputs. Universal Serial Bus (USB) is a type of Plug and Play connection that allows to connect a I/O device without shutdown the computer.

The IEEE-1394 multimedia connection is the fastest transfer device with no image degradation, also known as *iLink* or *FireWire* in reference to its speeds of operation (transmission rates can go up to 800MB). It provides a simple, low-cost, high-bandwidth isochronous (real-time) data transfer. It is ideal for digital video cameras and similar devices because of the mode of its transmission.

To construct a software that has to capture images from cameras or camcorders to a computer, we need to use APIs for these connections. APIs are Application Programming Interfaces that offer a standard development platform which allows all programs to write instructions for specialized hardware features without knowing hardware-specific code. On Windows platform this is provided by Microsoft DirectX Technology. In our context, we will not explore this technology because this is not our focus. Under Linux, where we are working, we use Video4Linux for Analog Camcorders and libraw1394 for DV Camcorders.

2.1. Image Capture and Exhibition in Linux

To capture images from devices on Linux, we need linux kernel's drivers, also called modules, and libraries to control our device.

Analog image capture is made by video4linux. Video4linux, also known as v4l, is the video capture API for the Linux kernel. The second generation of this API has already been released and is known as v4l2.

With a 1394-based digital camera we need ohci1394 (if we are using an OHCI compliant card), ieee1394 and raw1394 modules in the kernel space and in the user space, libraw1394 library provides the interface for the application programmer.

In general, we can see the linux 1394 subsystem [Linux1394 1999] like described in Figure 1.

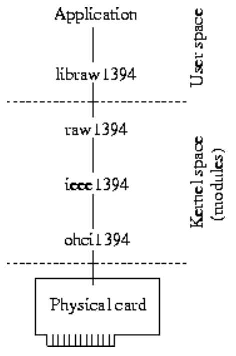


Figure 1. The 1394 hierarchy

Ieee1394 is the core of this subsystem, managing all high- and low-level drivers in the subsystem, handling transactions, and providing a mechanism for event triggering. Ohci1394 module is a low-level driver that interfaces the 1394 card and raw1394 is a high-level driver which provides an interface for userspace applications to access the raw 1394 bus.

3. System Architecture

The MUAN system was developed for Linux platform and the software part uses a lot of 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 comming from a camera video stream and allow user to select the desired images. A set of captured/selected images will form an animation. How this process works and its requirements are described below.

3.1. 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 Stop Motion because it allows to compare the last frame with the image coming from the camera (live incomming frame) so the user can check if the frame to be stored is the desired one and can adjust the scene accordinly. 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, as shown in Figure 2.

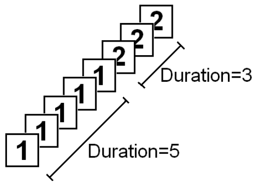


Figure 2. Representation of frame duration within the animation.

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, on the user interface we provide components for playing or stoping 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, user can choose if application is in 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 system color. But in the list, the stored frames are saved in DV format. Althoug this format contains image and sound information, its lenght is more compressed than a simple array of pixels.

Because the application has to show live frames comming from the camera, we create a thread which is continuously getting images from the camera in real time. When connected via firewire, application establish a communication with the camera and grabs

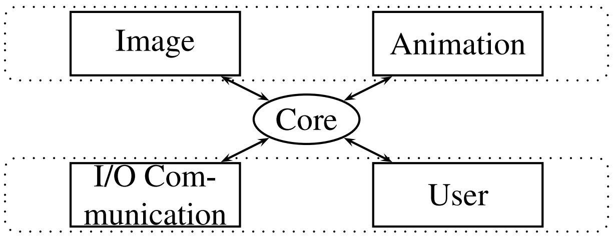
DV data. Images in DV format have 3:2 aspect ratio, that is, widescreen. In our purpose, we are not working with sound for the animation, we are focused on capture only images, so the captured DV frame has its sound track erased. When the connection with camera is over a composite or s-video input, application opens a direct communication with device, through video4linux, capturing images as an array of pixels in 24 bit RGB pallete. In this case, captured images are in 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 desired frames were captured, user can save the animation asking to application for 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 these files and edit it 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 a helpful feature for the animator to analize the last or next frame to be captured, are also present in user interface, leading application to be more friendly.

3.2. System Interface/Structure/Requirements

We can describe the interactions of our system like this:



In order to provide functionalities listed in section 3.1, we used some libraries commonly used on Linux platform and 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 ieee1394 (see section 2.1) was based on IEEE1394Reader class which is part of Kino [Schirmacher 2002] application and asynchronously grab DV data through the libraw1394. So, we create a buffer where the frames that are being continuously grab through a thread are stored. When application requests a frame, this library returns the last collected frame. In the case of v4l connection, video library interacts directly with v4l driver requesting an image every time the application needs.

To implement the functions that store the animation in video files, we use the libavcodec and libavformat libraries, which compose the FFmpeg Project [Bellard 2004]. FFmpeg has functionalities to record, convert and stream audio and video and is developed under Linux, but it can be compiled under most operating systems, including Windows. Libavcodec library contains all the FFmpeg audio/video encoders and decoders. Most of this codecs were developped from scratch to ensure best performances and high code reusability. Libavformat library contains parsers and generators for all common audio/video formats. With these libraries, application is able to read and write MPEG files.

To import JPEG images to an animation or export the frames of an animation to JPEG images, we use libjpeg version 6b [IJG 2001] which is a widely used free library for handling the JPEG image format, writed and distributed by IJG. Our code for converting to and from this format was based, respectively, on cjpeg and djpeg that are programs included in this library.

The class ImgList implements the animation representation with a list used to store frames. And to represent images or the frames of an animation, we used 2 classes: one that implements the frame only as an image, that is, an array of pixels (colors), and another that implements the frame as DV data, like it comes from digital devices through a firewire connection. This second representation was also implemented based on Kino application, using the Frame Class which provides utilities for processing digital video frames through the libdv. The Quasar DV codec [Krasic and Walthinsen 2003], or libdv, is a software codec for DV video, the encoding format used by most digital camcorders, typically those that support the IEEE 1394 interface. Libdv was developed according to the official standards for DV video: IEC 61834 and SMPTE 314M.

Another widely used free C library that we use in MUAN is libsndfile [Lopo 1999]. 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. On linux, libsndfile can works over ALSA driver (Advanced Linux Sound Architecture), which provides audio and MIDI functionality to this operating system, and we are requiring this option. As already said, we are not working with sound for animation, but sometimes we need sound on the user interface providing a better and easier interaction.

4. Applications

We developed two applications: Muan and Muan Player.

The first one allows users to create, edit, manipulate and play animations, acquiring images from a DV or analog cameras connected by firewire connection (via IEEE1394) or analog capture card installed (via v4l).

This application has two user interfaces: Muan AE and Muan AM. Muan AE was the first GUI constructed especially for AnimaEscola Project (see section 5.2) and is very user friendly. It consists of one unique main window that provides an image area, where images incoming from camera or captured images are displayed, and large buttons for each operation, making the application easier for kids and inexperienced (beginners) users that appear in the scope of AnimaEscola Project. This interface can be seen in Figure 3.

Muan AM, displayed in Figure 4, was a new GUI created to make the software more similar to the others graphic applications well established for the Linux platform. It has a main menu window, an image window and floating toolbars. The user can arrange the windows and dialogs according to the best position or desired style on the screen. This interface has its layout more directed to professional use.

Muan Player is a simple interface (Figure 5) for playing animations. From version 3.0, it was incorporated valuables resources of import/export animations from/for difer-

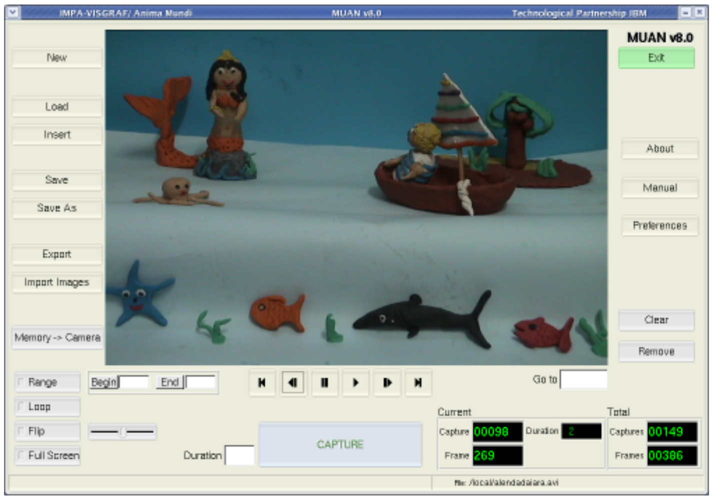


Figure 3. Muan AE interface.

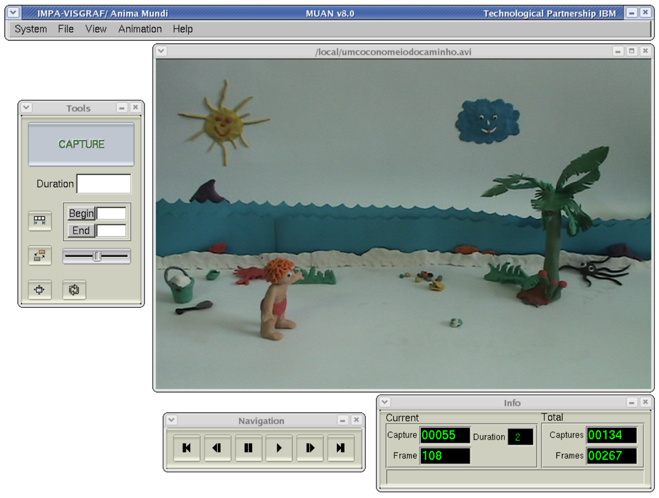


Figure 4. Muan AM interface.

ents formats and play animation in full screen. If user has a collection of images, he can create an animation from them, importing all JPG images from a directory (application uses numeric-alphabetic order) and then saving in desired video file format. Moreover, this application can be used on others platforms, since the ffmpeg library has been instaled.

5. User Experiments

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. The following subsections describe two successfull examples of the use of MUAN System.

5.1. Anima Mundi

Anima Mundi is the International Animation Festival of Brazil. Annually, since 1993, it presents competition screenings, special programs, retrospectives, international guests

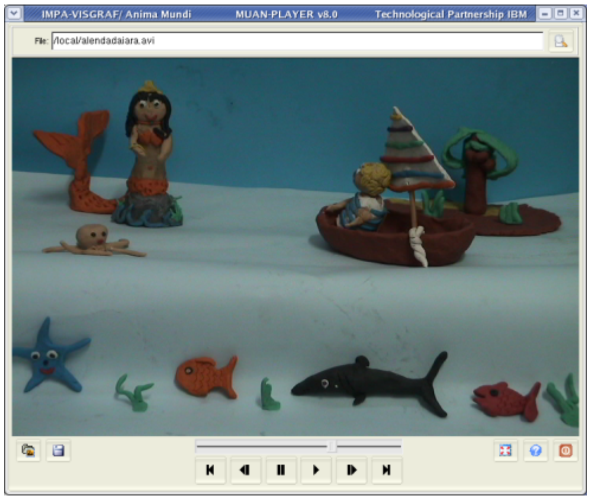

IMPA-VISGRAF/ Anima Mundi
MUAN-PLAYER v8.0
Technological Partnership IBM
File: /local/alendadaiara.avi

Figure 5. Muan Player interface.

and professional workshops. The MUAN System is used in the open and free workshops that allows the spectator to become the creator, producing animations from clay, drawing, sand or cut-outs.

5.2. Anima Escola

Anima Escola is a project that offers animation workshops for students and teachers from municipal schools of Rio de Janeiro in Brazil. Teachers and students can produce short animated videos using characters made with clay or using their own body (pixilation technique), advised by specialized professionals. The final results of this frame by frame video are seen immediately. The software MUAN for Linux integrates a tecnological system which allows the use of animation in schools. It is a valuable instrument contributing to improve the quality of public education. Through a creative use of avaiable technology, MUAN gives children an innovative and exciting learning experience.

6. Final Remarks

Currently, we are working on an open source version of MUAN software. The installation files (binary distributions and source code) and additional informations like documentation, manual, FAQ and install instructions will be available at http://www.visgraf.impa.br/muan.

We would like to acknowledge the support given by IBM with the special colaboration of Patrícia Menezes, Gabriela Villas Boas and Marco Aurélio e Souza.

References

Bellard, F. (2004). Ffmpeg multimedia system. http://ffmpeg.mplayerhq.hu/.

IJG (2001). Libjpeg. http://www.ijg.org/.

Krasic, C. and Walthinsen, E. (2003). Libdv. http://libdv.sourceforge.net/.

Linux1394 (1999). Ieee1394 for linux. http://www.linux1394.org/doc/overview.php.

Lopo, E. (1999). Libsndfile. http://www.mega-nerd.com/libsndfile/.

Schirmacher, A. (2002). Kino. http://www.kinodv.org/.