Ribbons: a live cinema instrument

Tomás Laurenzo
Universidad de la República, Uruguay

Abstract

Ribbons is a relatively small real time visual instrument created for maximizing the expressiveness of the performer within it's aesthetic paradigm. It's design questions the basic assumption of a flat rectangle as the traditional projection space by presenting a virtual three-dimensional space where the footage (or live video) can be projected and deconstructed, adding a new dimension of expressiveness orthogonal to traditional narrative.

We worked with the following design axes: playability vs. autonomy, expressiveness vs. narrative, and originality, and applied some of the basic techniques of Human-Computer Interaction and digital lutherie.

1 Introduction

1.1 Live cinema

Live cinema is a term recently coined for a long standing practice: real time audiovisual performances, which - in its current incarnation - are real time collaboration between sonic and visual artists (Makela 2006). Although the aesthetics of live cinema has been shaped mainly by VJing (the real time video mixing of footage) i.e. club-based visual performances - live cineastes have been performing at different spaces, with their oeuvres being shown in places ranging from traditional art galleries to multitudinous rock performances, and expanding traditional narrative cinema with a much broader conception of cinematographic space (Makela 2006).

This expansion, together with the images of club VJing, have led the production to very abstract and synaesthesia-focused works that somewhat deny traditional cinematographic narrative techniques and methods. This biases the production, by focusing only in 'the transitions, the movements, the pure visual beauty' (ibid). By claiming freedom from the narrative strings, the performer is not allowed to convey a potentially denser stream of images that benefits from less abstract images.
Live cinema’s performances, beyond their particular characteristics, are constructed by real time editing live or stored visual media (often both), using many gestures of traditional cinema (such as slow motion) and effects (such as scratching) that belong

to VJing. In order to permit these on-the-fly manipulations, different tools - both software and hardware - have appeared. The software tools range from the most general and low-level, for example Cycling74's Max/MSP/Jitter or Apple's Quartz Composer, which are full programming languages, albeit visual ones, to more application-like environments such as Resolume, Oscil8, etc. Hardware tools include video mixers, effects, and - of course - playback and output hardware.

1.2 Visual lutherie and Human-Computer Interaction (HCI).

These performance-oriented tools that produce moving images are called visual instruments, and therefore, their crafting should be called visual lutherie. As Miller Puckette said about computer music software: 'The design of the software cannot help but affect what computer music will sound like' (Puckette 2002), visual lutherie (as any tool used in art production does) influences visual production. Reciprocally, yet still talking about music, Bahn and Trueman present the concept of 'composed instruments' (Bahn 2001). If we believe 'new music tends to be the result of new techniques, which can be both compositional or instrumental' (Jordá 2005), we conclude that a possible approach for art production consists in the creation - the composition - of new tools of artistic performance, new instruments, and that it's creation may no longer be a stage previous to the art, but becomes part of it.
But this artistic approach to instrument creation should not forget that many guidelines and techniques of HCI are applicable (if consonant with the artist's desires) and aid in the instrument's design (Laurenzo 2008).

Two of the most important methodologies of HCI are user-centered design and iterative design, where the user becomes part of the development team - because he or she is an expert in his or her area of knowledge, and the team assumes that their work is perfectible and iterates by creating many versions that get closer and closer to what the user needs. Also, a very important interaction style is Direct Manipulation, which stands for interactive systems with continuous representation of the domain of interest, with rapid, reversible, incremental actions and continuous feedback. This allows the user to feel that he or she is operating directly with the objects presented to them with a direct representation of the domain of interest.
Both the methodologies and the interaction style are applicable to visual lutherie. In the following section we will present our live cinema instrument, which was created with these HCI concepts in mind.

2 The instrument

2.1 Design

Traditional cinema projects it's narrative onto the flat canvas of the projection screen: everything that happens in the film is under the 'tyranny-of-the- rectangle'. The live cineaste is also constricted by the same limitations, although it can be altered many times by using multiple projection screens which break the traditional rectangle, or by using projection mapping techniques.

However even in the most extreme cases, once the projection surface or surfaces have been chosen, all the narrative occurs on those pre-defined canvases. While *Ribbons*, like many visual instruments is at its core, a video player and is able to reproduce the videos in a standard way (a full-screen flat representation), and to apply some basic effects such as transparency, scratching and direct access, it's design challenges the flat representation by projecting the cinematic material (prerecorded or live) onto a three-dimensional, virtual, radically deformable canvas (see Figure 1).

To be able to do so, *Ribbons* creates a grid of three-dimensional particles with each one tinted with the corresponding colour of the video. The particles can be manipulated by the performer in novel ways, thus adding a new dimension of expression, orthogonal to the footages' original one, and distinct from the common VJing techniques.

This new dimension may or may not compete with the traditional one, and it is the performer's call to keep the images intelligible or completely deconstructed.
These particles can then be used as input for different visualizations (such as triangle trips, cubes or lines), which we shall discuss later.
In the design and construction of the instrument, three axes guided our work: playability vs. autonomy, expressiveness vs. narrative, and originality.

Figure 1. Different stages of deconstruction of the filmed image by applying a Perlin wind

2.2 Playability vs. autonomy

The defining characteristic of an instrument is that it is playable. Ideally, the user should feel that the manipulation is as direct as possible, even to the point that it's manipulation disappears from the cognitive universe of the user as he or she focus on the results: the interaction becomes a metaphor of a world instead of the metaphor of a conversation, that is, the manipulation is direct.

In order to reinforce the directness, all the commands built trigger an immediate response, and the user can directly control parameters (such as camera orientation), select representations, or set off some visual response (e.g. drawing text).

However, we wanted the instrument to be able to 'play by itself', that is, it should be able to keep on producing visual output even if the performer is not interacting with it. This was mainly because in real time performances sometimes one needs to focus on something else (e.g. a hardware video mixer) and the show must go on.
Two things were implemented to achieve this: sound reaction (the instrument processes the audio captured by the computer’s microphone and modifies the visual output) and inertial representation.

By inertial representation we mean that *Ribbons* allow the performer to deform the grid of particles by applying forces to them, and the particles act as if attached to strings (and then will oscillate and eventually converge to it's original position) creating an effect of deconstruction and reconstruction of the original frame that can be controlled by the performer. This allows the performer to deform the grid in such a way that it will keep on moving coherently, even if there is no user input with the synaesthesia reinforced by the before mentioned sound reaction.

The deformations can be completely random or coherently random (by using Perlin noise) and the performer can have medium to little control of each particle's actual movement but can always modify global parameters - like the strength of the strings, the direction of the particles, etc.

The final product is a visual instrument where the user can completely engage in the performance, yet is able to let the instrument perform by itself without the change being noticeable by the audience.

2.3 Expressiveness vs. narrative

As we mentioned, the performer can, for example, apply a Perlin 'wind' to the particles and deform the projection surface, even to the point of deconstructing the video frame, re-signifying its components (the pixels) as elements capable of independently conveying meaning.

This dichotomy between the narrative encapsulated on the cinematic material and the expressiveness of its manipulation conformed our second design axis.
Both the controllable deformations and the usage of the videos as raw data for the representations allow the performer to maintain the expressive language of traditional cinema while adding an orthogonal channel of information, expanding it for real time performance.

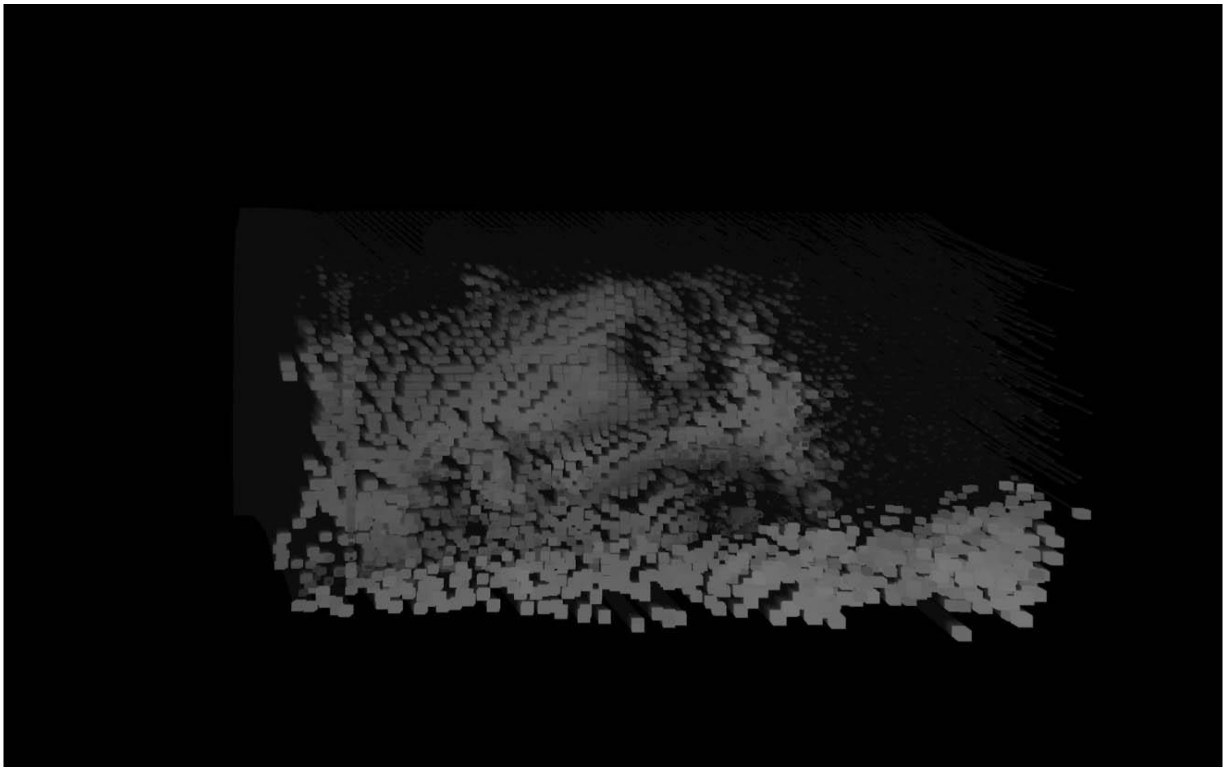

Figure 2. Ribbons screenshots: triangles (left) and triangles + lines (right) visualizations.

2.4 Originality

Our third and last axis of work simply consisted of the attempt to generate a distinct, recognizable visual output. Although we believe that we were moderately successful at this, we also coded some visualizations that are well known by the live cinema aficionado. For example, one of the completely sound-reactive outputs of *Ribbons* is directly inspired by, and reminiscent of, the visual output used by German artist Alva Noto in his latest tour. However it is obvious that the choices on whether or not use these visualizations, or how to combine them, is on the performer.

2.5 Operation.

Ribbons is controlled with one hand on the computer's keyboard and the other one on a drawing tablet (although it can be controlled with a standard computer mouse instead, the direct mapping from tablet-coordinates to screen-coordinates allows *Ribbons* to give an implicit feedback of the current level of the parameter being manipulated).

There are four different types of commands: *Selectors* select a video source or visualization with a keystroke; *Triggers* trigger an immediate visual response (such as drawing some text on screen or reversing the particles' rotation direction). Also with a keystroke (usually augmentable or modifiable using the shift key); *Faders* change a continuous value, such as rotation or return speeds. These are controlled by holding a key pressed and moving the pencil; and *Control commands* are metacommands (i.e. not belonging to a *Ribbons'* performance but commands for settings, quitting, saving, etc.).

2.6 Implementation

Ribbons was fully implemented in C++ and OpenGL using OpenFrameworks as a programming framework.

3 Conclusions and future work

We have shown our visual instrument *Ribbons,* which is not only theoretically consistent, but has also been successfully used in 'real life' performances (see Figure 3), where it provides the performer engagement that is expected from an instrument, while also being able to perform autonomously - delegating some decisions to an automated process, which also means to delegate some decisions to a previous ourselves (for brief periods).

The instrument allows us to investigate and question the basic need of expressive footage, and it's relation with, on one hand more abstract, generative visuals and on the other hand, it's real time manipulation by the performer. It also questions, by virtually projecting the footage onto a three-dimensional space where the camera can be moved around and the projected image can be deformed, the classic assumption of a flat orthogonal projection - without the costs and rigidity of more actual solutions. Finally we would like to praise frameworks such as OpenFrameworks and Processing, that allow for the creation of early prototypes quickly - offering artists the invaluable gesture of sketching.

Fig. 3

References

Bahn, C. and Trueman, D. "Interface: electronic chamber ensemble" in *ACM SIGCHI Workshop on New Interfaces for Musical Expression.* Electronic Proceedings, April 2001.

Jordá, S. *Digital Lutherie. Crafting musical computers for new musics' performance and improvisation*. Ph.D. thesis, Universitat Pompeu Fabra, 2005.

Laurenzo, T. "HCI in new media art practices", *Proceedings of Interacción 2008,* June 2008.

Makela, M. *Live Cinema, language and elements*. Masters Thesis, Helsinki University of Art and Design, May 2006.

Puckette, M. "Max at 17", *Computer Music Journal*, 2002.