to my parents

ABSTRACT

New media art constitutes a relevant field of artistic production that involves the use of new technologies in its production or presentation.

This master thesis examines the relationship between art and technology and proposes that the defining characteristic of new media art is media appropriation: the reconfiguration of technology as a medium instead of as a tool.

This work also examines the relationship between new media art and two important fields of computer engineering: human-computer interaction and computer programming, showing that they play a fundamental role in both media appropriation and in the construction of new media art's artistic language.

The thesis concludes by presenting three art objects that exemplify the previous discussion and where created in the context of this work: *Ribbons*, a visual instrument, *Puzzling* an art installation and *YARMI*, a musical instrument.

PREFACE

This is my Master Thesis on new media art, submitted to the program of Computer Science, within the Program of Basic Sciences (*Programa de Ciencias Básicas* - PEDECIBA) of the Universidad de la República, of Uruguay.

This fact –a thesis on art in a computer science environment– that some find natural is not inconsequential, for it is the first effort of this kind in our university.

The work that I am presenting here did actually start when I was an undergraduate student: then, thanks to Dr. Gregory Randall who was the director of the Electric Engineering Department (IEE[1]), I was able to work on digital lutherie for my undergraduate thesis.

Since I began working for my Masters' degree, I intended to study new media art and its relation with Human-Computer Interaction, and set my first framework in the construction of a Live Cinema instrument.

The work's goal was then to construct a visual instrument that allowed the artist (in this case called a VJ or visualist), to improvise over a previously defined, *scripted*, structure. The name of this first idea was *Visual Jazz*: *real-time scripted visual improvisation*.

I started working on this instrument, using Cycling'74's Max/MSP as my main programming language and different peripherals, among them Nintendo's *Wii Remote* and *Nunchuck*.

But many stories have prosaic condiments: when this work was well advanced, my house got robbed; I did lose my laptop and all my work. I have learnt my lesson and backup periodically, both locally and remotely.

So I had to start again and, this time, I realized that I did not want to focus on the creation of a product, but rather to work more at some theoretical aspects of new media art, human–computer interaction, digital lutherie, creative coding and their intersection.

And so I did. During this work I created some art objects and tools and was lucky enough to publish some papers.

I have also set the basis for more work that I am eager to tackle in the immediate future.

[1] *Instituto de Ingeniería Eléctrica*: Electric Engineering Institute. http://iie.fing.edu.uy/

Thesis contents

The main objective of this work is to introduce the reader to the new media art field of practice and research by focusing on its relationship with some fields of Computer Science: HCI and computer programming.

It also discusses two often overlooked aspects: how can Computer Science expertise help, influence and inspire art-making and what geographic considerations do new media artists should have.

The document begins with a discussion of what the new media art field is and its characteristics, where we[2] will focus on what we find to be –together with the use of new technologies– its most defining aspect: media appropriation.

In the next two chapters we will discuss the impact on art practice of two fields of Computer Science: Human-Computer Interaction and computer programming.

In chapter two, HCI, we will slightly focus on a digital lutherie as a case study that will end with a discussion on low cost lutherie as an alternative for third-world artists.

The following chapter will show some of the most used programming languages in the new media art field, focusing on what makes them successful and appropriate for the task.

In chapter five we will show three of our own artistic works, discussing their idea and implementation.

The sixth and last chapter shows the conclusions of this work and some possible future works for its author.

---

[2] During this work we will be using the first plural person, as we find it –probably because our Romanic roots– more conventional and impersonal.

ACKNOWLEDGEMENTS

This work would not have been possible without the help of several people.

First of all, I would like to thank the director of the research group I belong to, Eduardo Fernández, who was among the first to agree with me that media art; design and aesthetics might have their own space in our computer science lab, and thus allowed me to work in my areas of interest all these years.

I would also like to thank Drs. Eduardo Grampín and Sergi Jordà, my tutors, for their trust, help, support, and guidance. Likewise, I would like to thank this thesis' reviewer, Dr. Álvaro Cassinelli, whose insightful comments and suggestions were invaluable.

Also to be thanked is our librarian, Joseline Cortazzo, who has been of great help in finding books and papers.

I would like to thank, too, some of those many who helped me in different moments of my work: Marcelo Bertalmío, Javier Frank, Ramiro Kollmansperger, Marco Angulo, Óscar Angulo, Patricia Machado and, very specially, Silvia Viola.

Also very important to this work was the collaboration of my student Ernesto Rodríguez and my colleagues Juan Fabrizio Castro and Guillermo Moncecchi. Ernesto participated in *Puzzling* and *YARMI*, Juan participated in *YARMI* and in the *Tecnocordio*, and Guillermo took this thesis completion as a personal matter, not providing only valuable suggestions but also outstanding encouragement.

I would like also to thank all my fellow members of the *Taller de Computación Física Montevideo*, especially Enrique Aguerre, Luisa Pereira-Hors, Brian Mackern and Fabián Barros.

Finally, it is to be said that nothing would have been the same without María Eugenia Bellini's help, support, patience, and love.

INDEX

1 NEW MEDIA ART

But was it art? Was it engineering? Did this matter?

Johanna Drucker [32]

Introduction

Although it is commonly assumed that art itself lacks a satisfactory definition, being described as "one of the most elusive of the traditional problems of human culture" [140], much work has been done in aesthetics and art theory in order to develop a taxonomy for it, where different art forms are formally described.

Even if, at some level, it can feel contradictory to try to establish a classification on something whose essence has not been defined (and perhaps cannot be defined), much of the art production, art tools and, in general, art-related activities can be categorized into different (although often intersecting) classes.

Being both art theory and aesthetics fascinating areas of study, a serious discussion of what constitutes art, while tempting, exceeds the scope and intention of this work. However, being it a work on new media art theory and practice, we do need not only to specify what new media art stands for, but to discuss to some extent its essence and implications in both technological and artistic production

A first, yet rather trivial, definition would be that new media art is the application of new media within the arts, while a second, but equally trivial, would state that what characterizes it is the application of existing or new technologies to the art practice [30].

The Australia Council for the Arts defines it as follows:

> *new media art describes a process where existing, new and emerging technology is used by artists to create works that explore new modes of artistic expression, ranging from conceptual to virtual art, through performance to installation art and involve new media in both how the work (or a component of the work) is conceived and created as well as the way in which it is presented to an audience.*
>
> Australia Council for the Arts [6]

These attempts at defining new media art leave us with some questions, as we are immediately forced to ask what new media stands for and what makes it so important it as to be considered a new art form. Moreover, is it possible to classify a portion of the media and communication fields, as intrinsically new? Taking into account its dynamicity, how can we qualify something as new and pretend that that stands for more than a little while[3]?

This definition also raises our main concern: how new media art would be different of all other art forms that are in relation to technology. Furthermore, as we will see, technology has always played a defining role in art.

In spite of the soundness of this questioning, it is possible to identify a *qualitative* change in media production and consumption with the advent of the so-called digital age, where new ways of artistic production –together with the recontextualization of traditional art practices– do appear.

We will attempt to answer these questions in the rest of this section, starting with new media and then looking into the relationship of art and technology.

New media

> *Once we have surrendered our senses and nervous systems to the private manipulation of those who would try to benefit by taking a lease on our eyes and ears and nerves, we don't really have any rights left.*
>
> Marshal McLuhan [82]

While it has been said that, like art, new media lacks a formal definition, there are plenty of studies in different areas that now are considered part of or extremely related to it, in particular the digital production such as computer graphics, computer music, human-computer interaction, etc.

Lev Manovich[4], in his 2000's book, *The language of new media* [80], offers a definition of what new media are[5], also stating that the media

---

[3] We remember Wise's 1973 manifesto, Electronic Arts Intermix: At the Leading Edge of Art, where he reasonably stated that "all art was contemporary when it was created" [139].

[4] Lev Manovich, b. 1960, Moscow, "the world's most widely-read new media theorist", according to the program for X-Media Lab Singapore 2007 (http://www.xmedialab.com/index.php/main), is the author of Soft Cinema: Navigating the Database (The MIT Press, 2005), and The Language of New media (The MIT Press, 2001) which is hailed as "the most suggestive and broad ranging media history since Marshall McLuhan". Manovich is a Professor in Visual Arts Department, University of California at San Diego and the Director of the Software Studies Initiative at California Institute for Telecommunications and Information Technology (CALIT2).

revolution that the computer has started has, even though “we are just beginning to sense its initial effects”, a more profound impact than previous revolutions, such as the printing press or the photography, as it comprises “the shift of all of our culture to computer-mediated forms of production, distribution and communication”.

According to Manovich, new media are media that involves a computer in *any* of its stages, being their conception, production, communication or storage. So then, what are the characteristics that differentiate them from old, traditional media?

Manovich’s claim is almost true: the vast majority of new media production involves computers, but we must not forget that one of the first major exponents of new media was analog video (with video art still a strong art practice, although frequently involving computers). Yet, it is possible to find examples of new media art that do not involve a computer on any of their stages.

One such example is *Random Access*, by Korean artist Nam-June Paik. Paik "stuck more than fifty strips of audio tape to a wall and asked users to ‘play’ the segments by means of a play-back head that Paik had taken out of a reel-to-reel tape deck and wired to a pair of speakers” (see Figure 1)[96].

This deconstruction of the tape machine conforms a paradigmatic new media art object that recovers the aesthetic dimension of the tape machine’s working and creates not only an art object but also a tool for its public expression.

If, as Grahame Weinbren[6] said, "the digital revolution is a revolution of random access" [136], Nam-June Paik’s work does present a key feature of new and digital media without being digital.

---

[5] Although Manovich refers to “media” as a singular, in name of the coherence of this work we changed it to plural, favoring the singular “medium”. Quoting the Merriam-Webster dictionary, “The singular *media* and its plural *medias* seem to have originated in the field of advertising over 70 years ago; they are apparently still so used without stigma in that specialized field. In most other applications *media* is used as a plural of *medium*. The great popularity of the word in references to the agencies of mass communication is leading to the formation of a mass noun, construed as a singular. [...] This use is not as well established as the mass-noun use of *data* and is likely to incur criticism especially in writing.”

[6] Grahame Weinbren, (b. 1947 in Johannesburg, South Africa) is a pioneer of interactivity. He made films since the early 1970s, and has edited features, documentaries, music videos, and commercials. He has published widely on interactivity and cinema, and has lectured on interactivity and cinema throughout the world since 1982. He has made interactive cinema art-works since the early 1980s.

He studied in the University of London and the State University of New York in Buffalo. He lectured in philosophy, film and art history at California Institute of the Arts, the

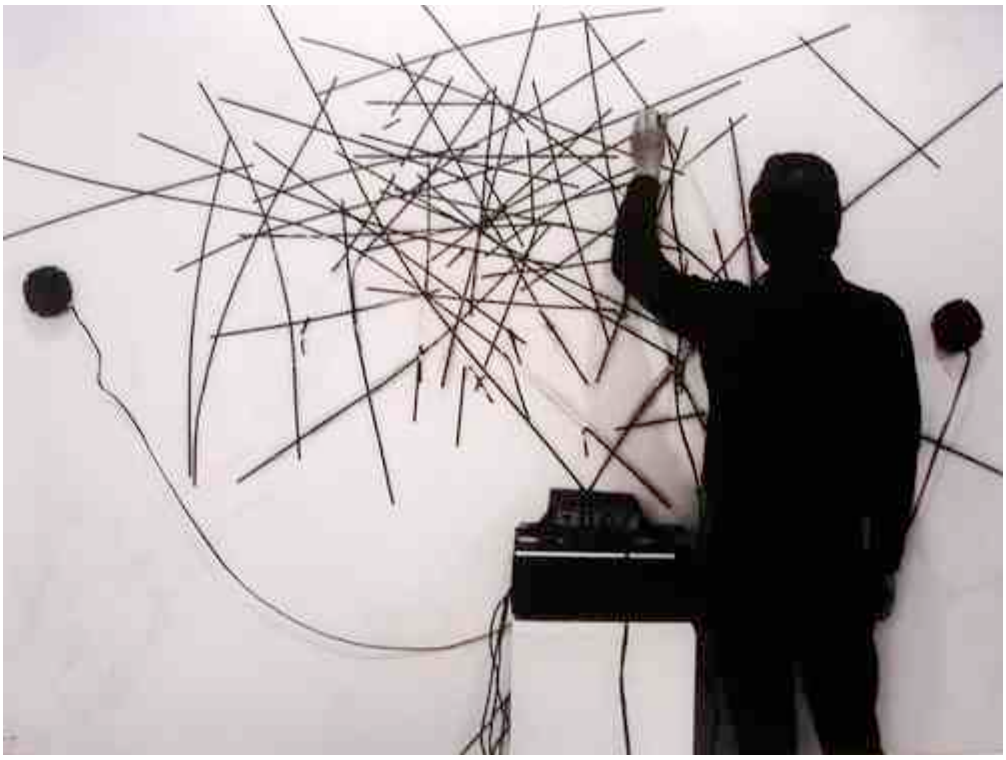

Figure 1 - *Random Access*. Nam-June Paik, 1963. Photography courtesy of Marc Wathieu, taken at *YOU_ser : Das Jahrhundert des Konsumenten* exhibition, ZKM, Karlsruhe.

According to Christiane Paul[7], the main themes of new media art do also surface in traditional media, and digital media "address issues that have been explored by artists throughout the centuries".

However, the appropriating way of new media is best exemplified by the digital.

University of California and the Hochschule für Bildende Künste in Braunschweig. He was Editor of the Millenium Film Journal, a journal of avant-garde film, video and works in other image technologies, New York; Since 1990 is visiting professor for photography und computer arts at the School of Visual Arts in New York City; Lives in New York City, USA.

[7] Christine Paul is Adjunct Curator of New Media Arts, Whitney Museum of American Art; Faculty, MFA Computer Arts Department, School of Visual Arts; and Director of Intelligent Agent, a print and online information resource dedicated to digital art. She has written extensively on new media, net art, information architecture and hypermedia; and her articles have been published in magazines such as Sculpture, Leonardo, and Intelligent Agent. She is the author of Digital Art (Thames and Hudson, 2003) and of Unreal City (Eastgate Systems, 1995). Paul teaches in the MFA computer arts department at the School of Visual Arts in New York and has lectured internationally on art and technology.

Manovich proposes five principles that summarize the differences between old and new media. As happens with almost all classification of human production, not all media artifacts obey these principles, so "they should be considered not as some absolute laws but rather as general tendencies" of this cultural production.

This characterization of new media (as a social, massive, phenomena) does not always translate linearly into new media *art*. As we said, many new media art's aspects and themes have been present in art for a long time, even before the appearance of the digital. Somehow reciprocally, new media art provides different insights into new media's communicational processes, offering alternatives that may not be in consonance with mass-media productions.

The five principles proposed by Manovich are: numerical representation, modularity, automation, variability and transcoding.

These principles –defining characteristics of new media– are, from the point of view of a Computer Science savvy reader, almost natural, even to the point that is really hard to think of a digital object that do not obey them.

Is it sound to consider them relevant, defining, while also natural and ubiquitous in digital domain? Would not that imply that new media production is merely a translation effort, a digitization?

In other words: do the characteristics depicted by these principles allow for new ways of communication? Or, is there something like an original expressive power of new technologies?

What we find (we will come back to this later on this chapter) is that new media artists and communicators have developed a *new language* that separates them from traditional communication, while many times maintaining the object to be communicated somehow intact.

We will now present and briefly[8] discuss Manovich's principles.

Numerical representation

The basic principle, the fundamental determining quality of new media, is that all its objects have a digital representation for which the source of information can be also digital or converted from analog sources.

This construction of a *representation* of the original information allows to manipulate it and to formally depict the manipulation. In Manovich's

---

[8] We will make emphasis in our view of principles proposed by Manovich, the reader is strongly suggested to also read Manovich's own construction of them. It is also to note that our view is constructed having a rather strong Computer Science background.

words “media becomes programmable” and programmability is “the most fundamental new quality of new media, which has no historical precedent”.

But it is worth noticing that, again, automatic manipulation of media is not inherently linked to digital representation; what are radically new are its easiness, its accessibility and its unspecificity. Even though analog manipulation of, for example, electromagnetic waves can be found as early as late XIX century, (with Tesla’s experiments on electricity in 1891), the construction of an electromechanical device for data manipulation, until this formalization, was for a pre-given purpose.

The digital revolution is a revolution of freedom[9].

It is inevitable to discuss the semiotic aspect of digital (or any type, for that matter), representation. What are its limits, possibilities and differences with other, possibly continuous, representation paradigms? Does the loss of information that digitization implies matter[10] or we can disregard it as non-significant (as it may not have a perceptual significance)?

Roland Barthes[11] stated that human communication is inherently discrete: “language is, as it were, that which *divides* reality” [11], while Manovich proposes that discretization is the natural conclusion of the serialization and standardization that the Industrial Revolution brought, both taking digitization as a natural human process of data representation.

In the context of this work it is enough to address that this is one of the basic question of semiology: the social construction and use of symbols; and that, from our perspective, it suffices to say that the artistic language that new media artists have constructed, and still are constructing, has its own characteristics and power.

---

[9] It is arguable that for people to be able to exercise that freedom it is needed to have the economic, social, *means* to the media. Freedom is not free. This will be discussed later in this chapter.

[10] A prosaic attempt to answer this can be found in [65]: “although time is mathematically different for everyone –depending on their velocity compared to the speed of light– the difference is so small that we can, and do, act as though time is constant. So we can say, as in deconstruction, that everyone sees a work differently, but the differences are small enough that viewers get something similar out of the piece”.

[11] Roland Barthes (November 12, 1915 – March 25, 1980) was a French literary critic, literary and social theorist, philosopher, and semiotician. Barthes' work extended over many fields and he influenced the development of schools of theory including structuralism, semiotics, existentialism, Marxism and post-structuralism.

It is also interesting that Manovich does not explicitly address the frontier between the digital and the analog work, which is traversed one and again by information. It is easy to spot data flows that trespass this border, almost seamlessly, many times. There are many examples of this, but one can be a musical instrument such as a guitar, played by a human musician, then digitized[12], transduced onto digital form. Then its digital representation is manipulated and, finally, sent back to analog form, for example as a visual or aural representation, in order to be perceived.

The importance of the frontier as a *cognitively tangible* boundary of the media object is to be addressed as it has a big impact in the design of the digital artifacts that are manipulated by a human user.

Modularity

This second characteristic of new media is, again, well-known to computer scientists: media objects are formed by an iterative semantic construction that starts from the bit and constructs more and more symbolic objects. This way, a three dimensional scene such as those often present in video games is constructed of several independent objects (pixels, polygons, textures, characters, scripts, etc.) that can be, and in fact are, re-articulated into different and heterogeneous media objects.

This modularity is present in all new media objects, as it is inherent to digital representation, playing a fundamental role in its communicative power.

Whether this principle is not an inevitable consequence of digital representation is debatable, but its role is undoubtedly important as a language-defining element with some artists explicitly using it as a style-defining characteristic. That is the case of remixes and mash-ups that combine different sources in order to obtain a new object[13]. Figure 2 shows an example of a mash-up by this work's author.

Automation

The third principle, and probably the one that encompasses the most radical shift of paradigm, is also a consequence of having a formal representation of media, and has to do with the *possibility* of automation of any of the different phases of new media creation and representation.

---

[12] Digitization of a continuous signal implies two phases: sampling (reading the analog signals at regular time intervals) and quantization (the approximation to a continuous range of values from the sampled data).

[13] A particularly famous example of mash-up is Danger Mouse's Grey Album, which combines music from The Beatles' White Album and Jay-Z's Black Album [48], being the musical mash-up that gained more notoriety.

The impact of the delegation of some (or all) the cognitive process that involves new media, is hard to conceive at once, and raises fundamental questions at the aesthetics and semiotic aspects of new media and new media art.

It should also be noted that the concept of delegation inevitably suggests an (usually impossible, as new media production almost always requires some machine's intervention) alternative process *without* delegation. That is, an analog process entirely performed by a person or a group of people.

Figure 2 - *Frida Klimt With No Monkeys Yet*. Tomas Laurenzo, 2007. A mash-up of Frida Kahlo's *Self portrait with a monkey* and Gustav Klimt's *Judith and the Head of Holofernes*.

To characterize the automation of the process as delegation, although sound from a historic point of view, is not always applicable. However, again it suffices us to perceive that some of the processes involved may belong only to new media realm. If there are no possible analogies with a traditional procedure, then there is no substitution.

What is also true is that a new compositional paradigm emerges, –one that we shall call collaborative– where a part of the cognitive work is done (directly or indirectly) by humans and some entirely by automated processes.

The automation is classified by Manovich[14] into two groups: *low level automation,* where direct manipulation of the data is performed by the computer, as when a filter to a signal is applied, and *high level automation*, where the computer (in this case, a more or less refined artificial intelligence system) must have some knowledge of the domain's semantics in order to perform the manipulation of the information. An example of the latter would be computer-controlled players in a video game.

This domain of knowledge can also enable computer systems to perform automatic tasks over vast amounts of information, addressing the problem of information retrieval (finding media), which, because of the over-proliferation of media sources and the easiness of storage, is increasingly important.

As we said, delegation does pose some fundamental questions on the artistic production: allowing an automatic system to take artistic or performative decisions do question the artist's role. If the artist were to only set the more general guidelines of an artistic performance to be actually executed by an autonomous system, both the role and identity of the artist becomes debatable.

Variability

Another principle of new media objects is their possibility to mutate, to adopt "different, potentially infinite, versions". It is easy to see old media products as fixed, and immutable, except for its natural degradation. Once again, the numeric representation of media and its modularity enables media producers (and, recently, consumers[15]) to generate new versions of old products.

Although it is true that new media is inherently variable (or, more correctly, the possibility of mutation is inherent to new media), many traditional media, especially in the arts, have, explicitly or implicitly, introduced chance and mutation as a part of them. We can see that a musical composition, written as a score, may be interpreted many times, each one lightly or dramatically different.

Variability is a natural consequence of digitization and modularity makes media manipulation feasible, and its impact in communication is huge. It allows producers to create, for example, new representations of the same set of data that are perceived as different products (this is well

[14] Manovich also talks about the "fractal structure of new media", a phrase that although does sound good, conveys a wrong concept as fractals are self-similar structures. New media, on the other hands is modular, but its parts usually are not structurally similar to its components.

[15] The neologism *prosumer* depicts users that are also creators of new media objects.

known in Human-Computer Interaction, where it is common to state that for users the product *is* its interface).

The relevance of the interface is absolute. For a media product to succeed, communication is needed. Semiotic theory tells us that interpretation is intrinsically tangled to the consumption of information, being both only one thing. There is no information without interpretation, and the latter depends on *how* the information is perceived.

This separation of media into "levels of content", data and interface, allows for a more richer, dynamic, and complex communication. An example of this would be hypertext, where both levels of data and the interface (which allows to acquire more information) are presented simultaneously, seamlessly integrated, and still completely identifiable.

Transcoding

After stating that new media (usually) lives as computer data, a new layer of interest appears: the *computer layer*.

The computer layer is the shape that information takes in the computer domain, together with the techniques that media producers utilize (always mediated by some kind of software) to store and manipulate the information.

This computer layer stores data that is relevant to, and in relation with, the *cultural layer*: the domain where the media product *lives* in the world.

In Manovich words: "we may expect that the computer layer will affect the cultural layer", that is, the essential characteristic of the media object. While it is common sense that the tools utilized do impact in the characteristics of the media object, we still need a more precise characterization of how the computer layer affects media.

Moreover, the cultural precepts of how a user interacts with a digital product have been adopted in function of what the user needs, but also, in correspondence of the computer (scientific) capabilities and methodologies of modeling the reality.

The computer layer itself is not something fixed but, on the contrary, it can change over time, once or multiple times, creating new environments where to live in, looking for its consumption.

Revising new media art's definition

Having a more precise idea of what new media means, we can take a closer look into new media art.

As we have already said, the definition of new media art can be somehow sidestepped by saying that it is art were new technologies are used by artists[16]. A better definition is given by Mark Tribe in his book aptly titled “New Media Art”, who says “we use the term new media art to describe projects that make use of emerging media technologies and are concerned with the cultural, political, and aesthetic possibilities of these tools” [127].

Yet again, Tribe tangles overmuch new media art’s language with its computer layer. What really happens is that new media art may or may not be digital, but the digital revolution has provided a *natural way* for it.

However, we prefer this second attempt at defining (or characterizing) new media art, for it addresses the interest of new media artists in the different possibilities of the tools utilized. If new media artists were using new technologies to re-create old-media art processes, there would be no fundamental change but, instead, an adaptation of known semiotic solutions, not posing new questions or problems.

The question is whether artists use technology as a tool or as a medium.

Reality shows a complex and rich landscape, partially due to the process that Pierre Lévy refers to as *virtualization*, which is defined as a dynamic that leads to a "...change in identity, a displacement of the center of ontological gravity of the object being considered". In this sense, virtualization is not strictly a technological change, or merely an epistemic reordering of cultural perception; rather, it amounts to a fundamental ontological shift in our very grounding in what we call reality [49] [128].

Virtualization consists in an “exponentiation of the entity under consideration. It is not a derealization (the transformation of a reality into a collection of possibles) but a change of identity, a displacement of the center of ontological gravity of the object considered. [...] Virtualization fluidizes existing distinctions, augments the degrees of freedom involved, and hollows out a compelling vacuum". Such a vacuum induces an "act of questioning" that Lévy says "is accompanied by a strange mental tension unknown to animals. This active hollow, this seminal void, is the very essence of the virtual".

This questioning of the ontological characteristics re-signifies the importance of the medium; in consonance with McLuhan’s “the medium is the message” it allows for its appropriation –one of the

[16] Quoting Bernard Baars: “one time-honored strategy in science is to side-step philosophical issues for a time by focusing on empirically decidable ones, in the hope that eventually, new scientific insights may cast some light on the perennial philosophical concerns” [9].

fundamental characteristics of new media art– and increases the degrees of freedom, catalyzing the ontological shift.

Appropriation plays a fundamental role in what defines new media art as an art genre on its own, it enables artists to search for (and to find) an artistic language that separates itself from the canons of mass media productions, from the intended use of the technology involved and also from traditional art forms.

Media appropriation.

> *Artists have always influenced and imitated one another, but in the twentieth century various forms of appropriation, from collage to sampling, emerged as an alternative to ex nihilo creativity. Enabled by technologies of mechanical reproduction, artists began to use found images and sounds in their work. Hannah Höch's Dadaist photomontages, Marcel Duchamp's readymades, Andy Warhol's Pop art* Brillo Boxes*, Bruce Connor's Found Footage films, and Sherrie Levine's Neo-conceptual remakes all reflected the changing status of artistic originality in the face of mass-produced culture.*
>
> Mark Tribe [127]

As Tribe also states, “in new media art, appropriation has become so common that it is almost taken for granted” [127]. Appropriation is often seen as a natural consequence of the use of the technology (in its natural form) directed towards other artists’ words.

In effect, the re-contextualization of artistic objects (being images, sounds, music, etc.), perhaps best epitomized by Duchamp’s works *Fountain* and *L.H.O.O.Q.* (see Figure 3) have played a major role in the artistic production of the 20th century. But, while artistic appropriation (sometimes considered an important post-modern movement [69]) still produces a big impact on the society, it is a somehow well-known artistic practice that remained unchanged in spirit until recently.

Appropriation calls into question originality, the romantic notion of the author, and other social institutions, like the gallery and the museum.

The artistic practices of appropriation artists, which often involve copying images of earlier art works, popular media, or advertising, come into conflict with copyright law. A good example of this would be Jeff Koon lost trials for copyright infringement [69].

These conflicts, as we will see, are deepened by new media art practices.

The ready-mades, or found art (everyday objects which have been taken out of their context and placed on display as art in an art environment, i.e., a gallery, museum, artist studio, etc. [50]) are some of the most radically appropriated objects, as they are (almost) not manipulated when re-contextualized (hence, "found art").

This artistic practice implied a radical shift from object to concept, in Duchamp's words a separation from "retinal art", with which he refers to the "interpretation of the visual world", which was an embrace of what became known as "conceptual art" [50].

Appropriation artists focused on the re-signification of already finished products, created by others. It is true that those products could have been anything, from classic artworks (as in *L.H.O.O.Q.*) to everyday objects (as the glass of water in *Oak Tree* by Michael Craig-Martin), but they were, always, finished products.

New media art, with its "intellectual parameters escaping disciplinary boundaries, asserting principles as much aesthetic as technical" [32], constitutes a fertile field for a new kind of appropriation, which operates on the processes of production instead of only on the final products.

Figure 3 - *L.H.O.O.Q.* Marcel Duchamp, 1919. It consisted of a cheap postcard reproduction of Leonardo da Vinci's *La Gioconda* onto which Duchamp drew a mustache and beard in pencil and appended the title. Duchamp (rapidly followed by other Dada artists) originated the ready-mades, appropriation art predates him.

This appropriation of the processes, which we call "media appropriation", is the differential characteristic of new media art, as it allows for the symbiotic relationship between art, technology and science, not only blurring their boundaries but –as almost every writing on new media art states– permitting their cross-fertilization and empowerment.

But one can reasonably say that, for example, many of Nam-June Paik's video installations –a paradigmatic example of early new media artworks[17]– are good examples of final product re-contextualizations which are classified as new media art.

Nam-June's Paik recontextualization, however, was a media appropriation as he re-converted the TV sets from passive communicators of information created elsewhere to active conveyors of meaning. The TVs became his appropriated medium.

However, after the massive instauration of new media products into everyday life, there is no possible re-contextualization as they are an assumed presence. Because of this, artworks such as Paik's *The more, the better* (a "media tower" comprised by 1003 monitors) would operate as traditional ready-mades (an artistic objectification of everyday objects) but not anymore as a *subversive use* of media products.

Another consequence of media appropriation is the utilization of technology as a raw medium of artistic production [68]; where artists must (and can, thanks to the modularity of new media) dip arbitrarily down into the technological components, adapt them, and create new ones, producing new new media artworks.

This operates not only as an artistic appropriation of the digital, but also as a catapulting force that liberates the artist from the constraints imposed by the available tools of media manipulation. It allows them to create a new artistic language that truly belongs to new media art.

New media art's language

George Dickie[18]'s *Institutional Theory of Art* [27], claims that the art status of a piece depends on the context in which the work is placed or

[17] We have chosen Nam June-Paik's works as an example not only because he was a famous new media artist but also because video art has been in the early years of new media art the most visible practice. Also, many of his works do have media appropriation; an example of it would be his first "satellite installation" *Good Morning Mr. Orwell*.

[18] George Dickie (b. 1926, U.S.A.) is a Professor Emeritus of Philosophy at University of Illinois at Chicago and one of the most influential philosophers of art working in the analytical tradition. One of his more influential works is "The Century of Taste," an inquiry into several eighteenth-century philosophers' treatments of the subject. The bulk of the work is devoted to championing, in a most forthright way, Hume's treatment of the subject over that of Kant.

viewed, while Arthur Danto[19]' [24] asserts that a piece's art status is dependent on the context and it's relation to the time and environment in which it was made [50].

> *To see something as art requires something the eye cannot descry – an atmosphere of artistic theory, a knowledge of the history of art: an artworld [50].*
>
> *What in the end makes the difference between a Brillo box and a work of art consisting of a Brillo box is a certain theory of art. It is the theory that takes it up into the world of art, and keeps it from collapsing into the real object that it is (in a sense of* is *other than artistic identification) [24].*
>
> Arthur C. Danto

The artworld is this context (social, political, geographical, chronological, etc.), and conforms the framework from which the art object is perceived, consumed, and judged. It includes the state-of-the-art artistic theory, and *evaluates*[20] art from this referential frame.

This evaluation can only exist if the artworld and the art object share some precepts, that is, if the art piece is *constructed* by using an artistic language that is somehow intelligible for the artworld.

Unlike many artistic movements, like Dadaism or Surrealism, that had a sudden, disruptive, appearance into the artistic landscape, and after being heavily criticized were co-opted by the artworld, the adoption of new media art has been slow, and its presence in the art museums circuit and art market was consolidated as late as 2000.

---

[19] Arthur Danto (b. 1924, U.S.A.) is currently the Johnsonian Professor of Philosophy Emeritus at University of Columbia. He is best known for his work in philosophical aesthetics and philosophy of history. He has been the recipient of many fellowships and grants including two Guggenheims, ACLS, and Fulbright. Professor Danto has served as Vice-President and President of the American Philosophical Association, as well as President of the American Society for Aesthetics. He is the author of numerous books, including *Nietzsche as Philosopher*, *Mysticism and Morality*, *The Transfiguration of the Commonplace*, *Narration and Knowledge*, *Connections to the World: The Basic Concepts of Philosophy*, and *Encounters and Reflections: Art in the Historical Present*, a collection of art criticism which won the National Book Critics Circle Prize for Criticism, 1990. His most recent book is *Embodied Meanings: Critical Essays and Aesthetic Meditations*. Art critic for *The Nation*, he has also published numerous articles in other journals. In addition, he is an editor of the *Journal of Philosophy* and consulting editor for various other publications.

[20] It is clear that the artworld changes through time, thanks to the driving force of new artworks.

This phenomenon can be in part explained by the absence of established, explicit, "manifesto-based", movements in contemporary art where individualism and diversity prevail, but also because of the intrinsic difficulty of media appropriation.

For new media to become part of the artworld, the artworld itself must develop a technically literate language of criticism.

It is intriguing that an art form that involves new media, a cultural product inherently massive and ubiquitous, had to face that much resistance from both the artistic and, to a lesser extent, technological fields; if a keen interest was to be found in technicians and scientists (although often biased towards the entertainment industry), the artists seemed to see new media art as a passing, shallow trend.

According to Hervé Fischer[21], this resistance climaxed after the dawn of avant-garde, which left us facing a crisis where novelty has no intrinsic value, not being anymore a characteristic to look for [40].

In effect, the avant-garde implicitly relied on the idea of "evolution of art": new art forms are better, art naturally improves; a concept no longer sustainable.

Although new media art's value does not come from its novelty or trendiness, both did collaborate to its appeal and consolidation: there have always been artists speaking about their time with their time's language, icons and symbols.

Besides media and symbolic appropriation, digital technologies objectively allow for new, unprecedented possibilities, multimedia, post-geography, and interactivity.

Multimedia is not new[22]. The simultaneous sensorial excitation, explicitly synaesthetic, has had its place in pre-digital art forms such as

---

[21] Artist-philosopher, Hervé Fischer graduated from the École Normale Supérieure, Paris. For many years he taught sociology of communication and culture at the Sorbonne. He obtained its MBA in philosophy and PhD. in sociology. He was a special guest at the Venice Biennial in 1976, the Sao Paulo Biennial in 1981, and Documenta 7 in Kassel (Germany) in 1982. He had personal exhibitions at the Musée Galliéra in Contemporary Art in 1976, at the Museum of Contemporary Art in Montreal in 1980 (a retrospective), and Mexico in 1983. In 1985, he organized Franco-Canadian participation in the Marco Polo electronic novel project which involved writers from Africa, Europe and Canada. Since 1999, he is working again as a painter of the digital age. He had one man shows at the Museo Nacional de Bellas Artes in Buenos Aires, Argentine, (2003), Museo Nacional de Artes Visuales, Montevideo, Uruguay (2004), Museo Nacional de Bellas Artes de Chile, Santiago (2006).

[22] Aside of explicitly multimedia works, such as Richar Wagner's *gesamtkunstwerk* –the "total artwork"–, Fischer said that art has always been multimedia, as -for example- Goya's "*El tres de mayo de 1808 en Madrid*" can effectively convey the sound of a shot.

theatre, opera and music. Nevertheless, new media art pieces can explore it in new ways, and question the representation of information in a conceptual-art way that differs from previous works.

Figure 4 - *Brillo,* Andy Warhol, 1964. Unlike the corrugated cardboard originals represented, these sculptures are made of wood.

What perhaps belongs entirely to new media art is post-geography, that is, geographically distributed, real-time communicated art objects. This communication can happen inside the artwork, between performers and public, between the art piece and the public, etc.

The third aspect we have mentioned is interactivity.

One can argue that art has always been interactive, for its consumption always implies a cognitive process from the public. However, new media art is *explicit* in its interactivity, and because of it, the cognitive task that art consumption requires sometimes shifts from the public to the artist.

Robert Browning's aphorism "less is more", is extremely contradicted by the exuberance of digital art, and –as a byproduct of this exuberance– the semiotic density decays. This makes digital art pieces to be perceived as shallower, or even colder.

Often its coldness comes from its explicitness.

However, it is clear that this is not a *condicio sine qua non* of interactive art. And is also true that low quality art has always existed[23]. What this does imply is that it is needed to find a new language that allows artists to express themselves and their culture, and also one where the artists expressive capabilities are expanded by the *possibility* of using it.

The much mentioned crisis of contemporary art, even to the point of assuring that art has ended[24] [25] is, then, a crisis of creativity that can be seen in the struggle of new media artists to define their language and to identify the cultural icons (technological, social, cultural) that are to be appropriated, rearticulated, used, by them.

It may not be an easy task; for example, new media artists are sill looking for ways of creating pieces that do catch the audience while not being ludic. This can be a particularly hard task, as new, interactive technologies are commonly and easily associated with entertainment and the entertainment industry (that is, the language that many times is assumed by the public is one ludic, of entertainment).

This struggle for a new language is hardened by the speed of technological advancement that makes new media art –which could speak about our time with our time's language– "look less anticipatory" [45], shifting the interest from the art expression to the technology itself.

As an attempt to aid to the comprehension of this relationship between (new media) art and technology we will take a closer look to some its history[25].

Art and technology

The interaction between art and technology is not a new phenomena, but has always accompanied the history of both art and technology during the history of mankind ("only with the invention of oil painting it was possible to paint outdoors, only the acrylic paint created the smooth surfaces that Pop Art needed" [46])

However, this relation acquired a new characteristic, a qualitative difference thanks to the unspecificity of the computer, both the

[23] Although it seems that extremely low quality art is easier than ever to create with new media tools, what really happens is that the potential-of-exposure of new media art pieces has increased dramatically in the latest years thanks to the consolidation of the field.

[24] It is obviously false to say that people do not paint, sculpt, or in other ways create art objects anymore, and it is not likely that they will stop doing so in the imaginable future. However, due to the perceived lack of substance, stating that art has ended means that it would continue as mere decoration or as entertainment [16].

[25] A complete recall of art and technology history exceeds the scope of this work, the reader can refer to [96], [127] and [114].

opportunity and curse of digital art, where everything is possible, but nothing necessary [46].

The relationship between technological or scientific advancements and new media art is so strong that the simplest way to trace a history of new media in art would be through the development of the technology itself. This view, although chronologically sensible, would not take into account the very important artistic developments that shaped new media art's language.

However, it is also possible to see it as a result of the studies on image and movement of the Futurists, or of the experimentation with radically new art forms that Dadaism proposed.

In effect, many Dadaist strategies reappear in new media art, including photomontage, collage, ready-mades, political action, and performance [127], and it is very clear that Marcel Duchamp (among Cage, Man Ray, Warhol and many others) prefigured many of the new media art concepts, works, ideas and tendencies.

> *How one feels about Marcel Duchamp is, essentially, how one feels about a great deal of contemporary art.*
>
> Michael Rush [114]

Accordingly, as we have seen, pop art and conceptual art can be easily cited as two important precursors of new media art, where the first one's focus in the "recycling of everything" is comparable to new media art's media appropriation, and the second one's focus in concepts over form is deeply related to new media's transcoding principle and its own focus on ideas and information.

It should be clear that the technology advancements played an *enabling role* in art, but, in consonance with conceptual art, also helped artists to delegate more and more of the creation to technological artifacts.

Herbert W. Franke[26], in his book *Computer Graphics - Computer Art* [44], proposes a classification of artistic production based on the degree of autonomy of the digital system used in art creation.

This way of seeing digital art's evolution explicitly puts the artist in its traditional role of consumer of technology, which, as we know, is being put under question by new media artists.

---

[26] Herbert W. Franke (b. 1927 in Vienna), Franke worked from 1973 - 1997 at the University of Munich, lecturing in computer graphics and computer art. He has also written widely on computer art and his first book *Computer Graphics - Computer Art* was the earliest comprehensive text on the subject.

Franke identifies four stages of evolution of the delegation of the aesthetic-creative process (see Figure 5), which gives us a compact, although reductionist, taxonomy.

This stages can be found in different art practices, being music the most evident, because sounds can be produced with very simple mechanical systems, while visual language is less mechanizable, and was deferred until the apparition of photography.

How apparent a delegation is, or how explicit the importance of the tools used by the artist are, depends heavily on the art field, as the following quote from Ken Perlin shows.

> *Oil painters use a controlled random process (centuries before John Cage made such a big deal about it).*
>
> Ken Perlin [99].

But new media art subverts this process as it does not intend to translate traditional art into the digital realm but, again by media appropriation, obtains a new field where the artworld's premises need to find a new conceptual frame.

This need of adaptation from the artworld appears evident when we look at the process of consolidation of new media art.

	Information-psychological conditions		Physio-physical conditions
	Information-condition	Redundancy-condition	
Manual art	Man	Man	Man
Instrumental art	Man	Man	Physical machine
Deterministic computer art	Man	Classical automata	Physical machine
Stochastic computer art	Random number generator	Classical automata	Physical machine

Figure 5 - Delegation of aesthetic-creative processes to machines, according to Herbert W. Franke [44]

Consolidation of new media art

The resistance from both artists and academia to accept new media art as a legit, respectable, art form is apparent when we look at how long it took for the society to dedicate space and resources to it.

In effect, if we are to measure the sociologic relevance of any cultural field by the number and the importance of the cultural institutions devoted to it, we can say it took at least ten years for new media art to move from the underground to the mainstream [81]. Although the relevance of the museum in contemporary art has been put under scrutiny –Douglas Crimp[27] concludes that the museum is no longer the *default* location of contemporary art– and therefore underground art has a new relevance (contradicting its defining quality). It is also clear that the importance of cultural institutions in the acceptance of new media art works, such as video art and interactive installation art.

While it would be too narrowing to assume that the history of new media art is the history of its institutions, it is easy to see that some institutions did play a key role as a way for artists to gather, and as a place for sharing, learning, and acquiring the needed visibility.

These institutions started to appear in the 1970s (SIGGRAPH in 1974, Ars Electronica in 1979) and were consolidated in Europe in the latest years of the decade of 1980, and in USA ten years after that: in 1995 the universities and art schools started to include new media, and nowadays there are numerous supporting foundations –in many cases belonging to the governments– and many groups and institutions who focus in research and artistic production.

It has been argued that this delay –which can seem very strange, as USA has a very rich history on technology-based art, with the MIT hiring its first artist in the 40's, and with successful private endeavors such as Experiments in Art and Technology (EAT)– was a consequence of both the social crisis and the shift from art to advertisement that the U.S. suffered in the 1980s [89], yet we believe that the importance of neither the institutions' role and the artists' comfortable decisions should be underestimated.

As a result of this, during many years Japan and Europe were the most active places in new media art, being USA much more conservative, directing the needs of artists through video art and Net.Art, A kind of

[27] Douglas Crimp is one of the major figures in contemporary art criticism in the United States. He has published widely in this field, both in academic books and scholarly articles, and in journalism and social criticism. He is a recognized expert on post-war American art and sculpture and one of the foremost commentators on the condition of art within postmodernity. Among his classic essays on contemporary art are 'Pictures' (1979), 'On the Museum's Ruins' (1980), and 'The End of Painting' (1981), which, like many of his other essays, have been reprinted numerous times. His book On the Museum's Ruins (1993) is a key text on art in the late twentieth century. His distinctive approach to visual studies is grounded in an impressively sociological and social-historical understanding of the life of pictures (and performances) within contemporary society, which makes it accessible to researchers in the social sciences as well as in the arts and humanities.

production more similar to entertainment, something that USA's public was probably more keen on.

New technologies where, then, used by American artists by an easier adaptation that didn't require a profound knowledge of the medium itself, opting by a more traditional role of artists that consume technological tools.

This kind of attitude towards technology, seems to appear naturally in contexts more-or-less conservatives, and the recovering of USA was possible thanks to the legitimization that academia brought by creating careers, courses, and post-degrees in the area, together with the legitimization provided of the museums circuits. "Finally in 2001, both Whitney Museum in New York and San Francisco Museum of Modern art (SFMOMA) have mounted large survey exhibitions of new media art" [81].

It is important to notice that, although being (relatively) conservative, USA could adapt its academia and art circuit by investing seriously. This solution may seem impracticable in less wealthy environments. We will discuss this in the final part of this chapter, focusing in our country's current situation.

An example of the third world: new media art in Uruguay

By using the same metric of measuring in function of the cultural institutions, we find that in Uruguay, nowadays (2008), there is no institution[28] of this kind, and we believe that one can find the same conservative approach to new media art of the early '90s in USA, both from the institutions and the artists.

Consistent to it, the art that is produced (and legitimized) in Uruguay mainly does not include the usage of technology as a medium. It is possible to see an example of this in the last *Premio Nacional de Artes Visuales*[29], arguably the most important prize on visual arts of Uruguay.

For the 52nd edition of the Prize, several pieces created with classical techniques were selected, and also some that include new technologies: an installation (Comas), murals (Sicco, Ronela) and video art pieces (Sicco, Sastre, Santiago).

In any of this cases there is an appropriation of the medium. Instead, all the artists use the technology as a tool, a way of constructing. That way,

---

[28] It is worth mentioning that in the Computer Science Institute of the School of Engineering of the Universidad de la República (Uruguay), there exists a research group on HCI, Computer Graphics and new media art, which we belong to, and is taking its first steps in this direction.

[29] http://www.mec.gub.uy/cultura/Concursos/ArtesVisuales/ArtesVisuales.html

the video art pieces show a very high level of control of *the tool digital video* (especially those by Martín Sastre, who does question from within the product, the medium that sustains it, playing with the blurred frontier between art and entertainment), and its language, but does not propose, nor produce, alternative ways of using the medium.

These artists do not question their medium from its conception but play with the content in the context of standard video productions. We can say that they do not belong to the new media art field, because its conception does not take part of the process of exploration of new ways of artistic expression.

In the same way, the exhibited murals show a refined control of the manipulation of the digital image, but for producing printed graphical pieces, *classical* ones.

The lack of questioning of the medium itself is particularly obvious in the piece by Comas, which consists in a laptop computer that shows different pages of a PDF file designed by the artist. Under the installation appeared a sign (presumably added by the museum's staff) that warned, "Do NOT touch the computer" (sic), turning the computer into a mere image projector[30].

What we find in the systematic appearance of this kind of pieces, is that re-creating a traditional communicational method with the computer was not a choice –or a re-lecture of the media– but an expression of how the artists see the technology: as a tool that has its ways already defined.

Digital inclusion

The recovering of USA was based on two pillars: first, the assumption from the cultural agents (states, academics, museums, etc.), of the necessity of acting in that direction, and second, the different actors' economic and infrastructural possibilities of construction of knowledge.

Although it is possible to say that in lot of aspects Uruguay does have the needed knowledge (or the tools to construct it), the infrastructural and, above all, economical limitations, raise a hard problem when thinking about policies and concrete actions.

Something that we have essayed, and believe in its soundness, consists in using obsolete technology, rearticulated for artistic or educational purposes [68].

[30] This does not mean a critic, but a factual asseveration. We do not claim anything about Comas intentions, instead, we find that established Uruguayan artists are not yet using technology as a medium.

This approach is not new, in Eladio Dieste's words:[31], "Each problem [...] should be faced with a sort of ingenuity, [...] with an attitude humble and vigilant. It should be thought again, with the basic body of knowledge that is now the heritage of all men" [29], but assuming it implies a radical change of attitude.

For this, it is imperative to acknowledge the asymmetries of first and third world, together with the differences between the solutions that can be applied to our context and the ones created under other parameters. These differences are even more important when "as a consequence of the equivocated attitude of imaging a science and technology already done, that only wait for us to discover them, a blindness is created among us" [29].

For making new media art creation viable in our context it is essential to build hybrid teams that equally engage artists, technicians, engineers and scientists, who would share the construction of knowledge and the needed tools, and enable the creation of their own language.

The emergence of studies on new media art testifies the acknowledgement of the key cultural role that the digital plays in our global society [81]. However, in many aspects new media art is still missing in the artistic and technological production of Uruguay (we suspect that a similar situation might happen in other countries).

This proposed attitude towards technology and new media art does affect the artistic production: in a context where state-of-the-art technology cannot be taken as a granted commodity, the nature of the relationship between artists and technology changes. This makes media appropriation an almost unavoidable path, and strongly impacts the emerging artistic languages.

An example of this methodology can be found in out musical instrument, the *Tecnocordio*, which is briefly discussed in section *Low cost digital lutherie* in chapter 2, *Human-computer interaction*.

---

[31] Eladio Dieste (1917-2000) born in Artigas, Uruguay, got his degree from the Facultad de Ingeniería (UDELAR, 1943). He was *Miembro Correspondiente* of the Academia de Ciencias de la República Argentina, Ad Honorem Professor of the Facultad de Arquitectura of Montevideo and the Facultad de Arquitectura of Buenos Aires, *Miembro correspondiente* of the Academia de Bellas Artes Argentina, and Doctor "Honoris Causa" of the UDELAR.

2 HUMAN-COMPUTER INTERACTION

The creative act is not performed by the artist alone; the spectator brings the work in contact with the external world by deciphering and interpreting its inner qualifications and thus adds his contribution to the creative act.

Marcel Duchamp [33]

New media art, as we defined it in the preceding chapter, is art that appropriates new media, new and emerging technology and explores new modes of artistic expression.

This media appropriation only can happen when the artist has a fair knowledge of the technological aspects of the media involved, which usually means –as one can easily infer from Manovich's new media principles– a fair knowledge of computers and the digital realm.

To put it in the words of Nintendo's Shigeru Miyamoto[32],

another important element is a belief that creators are artists. At the same time, however, it's necessary for us creators to be engineers, because of the skill required for the creations.

Shigeru Miyamoto [115]

From the plethora of fields that new media artists may need to be solvent in –obviously depending on their interests– there are some far more common, those that are used on a wide range of art objects.

As a result of the nature of the technological media, electronics and computer programming appear as the main tools for its appropriation.

However, being interactive art objects one of the more popular trends of new media art, designing such an interaction appears as a new problem space, turning human-computer interaction knowledge into a particular relevant area of knowledge.

A definition of human-computer interaction (HCI), is given by the ACM's Special Interest Group on Computer-Human Interaction (SIGCHI):

---

[32] Miyamoto Shigeru, (宮本 茂, born, 1952 in Japan) is one of the most renowned video game designers. He is the creator of the Mario, Donkey Kong, The Legend of Zelda, Star Fox, Nintendogs, Wave Race, and Pikmin video game series for Nintendo game systems.

> *Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.*
>
> ACM's Special Interest Group on Computer-Human Interaction [51]

There are other strongly related fields: human factors and interaction design. While HCI usually focuses on the computer side of the interaction (as it its definition shows), human factors does the same on the characteristics of human beings that are applicable to the design of systems and devices of all kinds [52].

Finally, interaction design explicitly focuses on the interaction, that is, an interaction designer does not (or, rather, not only) designs a machine or evaluates the comfort or the ergonomics of a controller, but instead focuses on the interaction that emerges when using a system in a given context.

We believe that this approach is richer than the ones provided by human factors and HCI, but, being HCI the more commonly used way of referring to the area, we will use both HCI and interaction design interchangeably.

HCI and software engineering

When beginning to study interaction design, one immediately finds that the majority of the literature produced in the early years focuses on showing the importance and validity of the area (something similar to what we frequently find now in new media art literature).

This has proved been no longer necessary; when software companies assumed that, for users, the product *is* the interface, HCI found its natural way into the software creation process.

However, HCI's inherent multidisciplinarity still brings some problems and tension.

In effect, it is easy to observe that interaction design involves computer engineering, psychology, physical and psychical ergonomics, psychophysics, sociology and design, which means that it is needed to modify the software production processes in order to allow specialists of this areas to integrate their teams.

The assumption of the importance of HCI in the software development cycle is proved by the appearance of several normative references and standards that aim to offer process models that effectively take into account interaction in the software construction.

Some examples of these references are [121]:

- Draft International Standard (DIS) 13407, Human Centered Design process for interactive systems. International Standards Organization, Geneva, Switzerland, 1997.
- International Standard 9001, Quality Systems - Model for Quality Assurance in Design, Development, Production, Installation and Servicing. International Standards Organization, Geneva, Switzerland, 1987.
- Draft International Standard (DIS) 8402, *Quality Vocabulary*. International Standards Organization, Geneva, Switzerland, 1994.
- Draft ISO/IEC 14581-1, *Information Technology Evaluation of Software products General guide.*
- Draft International Standard (DIS) 9241-11, *Ergonomic Requirements for office work with visual display terminals, Part 11: Guidance on Usability*, International Standards Organization, Geneva, Switzerland, 1997.
- Draft International Standard (DIS) 14915 - Multimedia User Interface Design; Software Ergonomic Requirements

In addition, relevant documents containing accessibility guidelines are the Draft HFES/ANSI 200, Section 5: Accessibility, as well as the World Wide Web Consortium (W3C) Accessibility Guidelines. Finally, W3C-WAI (Web Accessibility Initiative) pursues standardization activities in the area of accessibility guidelines.

HCI and new media art

> *A reciprocal relationship can be created between the practices of art and science that preserves disciplinary distinctiveness while challenging all participants in the areas where their respective disciplines are weakest.*
>
> Piotr D. Adamczyk et al. [3]

As we mentioned in the preceding chapter, the relationship between art, technology, and science, is not new but as old as art itself, and this unavoidable relationship (which is frequently re-discovered in the new media art field, although artists have long held an integral relationship with technology [23]), impacts not only in the arts but also in the technology.

This cross-fertilization between art and technology is frequently seen in HCI and new media art.

In effect, interaction design's techniques are frequently –albeit many times unknowingly– used by new media artists, while, at the same time, HCI is one of the fields where its experts explicitly look for artful techniques and tools and user experiences to support creative engagement or to foster creativity [23] [13].

Media appropriation appears when artists adopt the methodologies and techniques of HCI (and, of course, other technological or scientific fields).

This, however, can occur *directly*, when the artist has the needed knowledge or technical mastery, or *indirectly*[33], when a multidisciplinary team (comprising artists and technologists or scientists) collaborates in the artistic production.

Both direct and indirect appropriation may allow for the emergence of "new Leonardos", as in the cautious words of Roger Malina:[34]

> *Perhaps in our lifetime we will see the emergence of "new Leonardos", creative individuals or teams who will not only develop a meaningful art for our times but also drive new agendas in science and stimulate technological innovation that addresses today's human needs.*
>
> Roger F. Malina [79]

We affirmed that HCI is particularly important for new media artists. It is so, first, because its inherent multidisciplinarity provides with existing models of collaboration with other disciplines, and second, because many of HCI's classic guidelines, heuristic and methodologies are directly applicable to new media art production.

Also, if one is to momentarily think of art as communication (of course this is reductionist and debatable) it seems natural to use interaction design as a conceptual framework for new media art works (allowing the artist to conceive more effective ways of communicating a message), in

---

[33] Other authors, like Coughlan and Johnson assert that media appropriation might not happen, and they allow for a scenario where the technologist and the artist are unconnected parties [23]. As we stated before, we sustain that this is not new media art but rather a virtualized traditional art practice.

[34] Roger Malina is an astrophysicist at the Laboratoire d'Astrophysique de Marseille CNRS in France and Executive Editor of the Leonardo publications circulated by MIT Press. He serves as Chairman of the Board of Leonardo, The International Society for the Arts, Sciences and Technology, and is the President of the Observatoire Leonardo des Arts et Technosciences in Paris. He is Co-chair of the International Advisory Board of the Inter-Society for the Electronic Arts, a member of the International Academy of Astronautics, and is currently an Osher Fellow at the Exploratorium. He is a member of the International Academy of Astronautics and co-chair of their Committee on Space Activities and Society. Since 1982 he has served as Executive Editor of the journal Leonardo. He writes on the relationship between the arts, sciences and technology

a similar way that rich knowledge of music theory might help a music composer's work.

However, one should not be fooled into thinking that a HCI-based approach is to be of any systematic help (Ira Greenberg's "happy coding mistakes" [47] –the insertion of randomness by the programmer's inability of foreseeing the code's outcome– offer a great counter-example), or into considering that HCI-evaluation techniques can be used to evaluate art. As Eric Paulos puts it "you can't evaluate what you can't evaluate" [97], or in other words, better interaction[35] does not translate into better art[36].

This basic conclusion has biased much of the literature on the convergence of HCI and new media art, where a great portion of it states and re-states that new media arts radical experimentation and subversion of human-computer interaction can be used as a driving force for research in interaction [56]. It is less common to find a discussion on how HCI can help an artist's work.

With Paulo's caveat in mind, we will now identify some of the most important techniques of HCI that are (or can be) appropriated by new media artists. We will then focus on digital lutherie, as it specifically creates tools for artist production while –at the same time– blurring the distinction between tools and artworks.

Some aspects of HCI relevant to (interactive) art production

One of the most important concepts of HCI is user-centered design (a concept introduced by Donald Norman's research team in the 1980s), which is both a design philosophy and a set of design processes that allow for end-users to influence on how a design takes shape [1].

The need to integrate users into the design process has its roots in the assumption of the differences between the designers of the system and its users.

In the context of art creation, however, things usually are not as simple as in product creation: the artists' concern about their audience's characteristics could be in any place between nonexistent and complete.

This kind of reasoning can be used to dismiss many, if not all, of HCI's concepts. For instance, artists could just not care about an interactive piece's usability or learnability, but it is also true that they might care

[35] Also, HCI's goal of identifying "good interaction" conveys the problem of determining what exactly "good" means. Different metrics focus on different qualities, including speed, accuracy, recall, enjoyment, etc. [13].

[36] Paulos' paper is aptly titled "HCI cannot be used to evaluate art".

(even if only to ignore it, or to purposely construct a piece with poor usability).

Our analogy with a composer's knowledge of musical theory still applies: the amount of knowledge composers have does not say anything about how their pieces will sound, but it may expand the spectrum of the composers' possibilities. In the same way, we believe that HCI knowledge may not only expand the spectrum of possibilities of interactive art but also offer artists a deeper and finer knowledge of the interaction the piece is proposing.

An interactive artist may then apply some of the techniques and procedures that are common in interaction design aiming to improve his creative process. Some[37] of these are:

Rough sketching, early prototyping and iterative design

Although these are not techniques as much as heuristics (and, as with many others, it is in part "only" the systematization of common sense), it is reasonable to explicitly acknowledge that sketching and testing early versions of the interaction proposal should streamline its creation process and maximize the probability that everything goes as planned (which, of course, may not be in consonance with the artist's desire).

Early prototyping and testing also allows the creator to have an early feedback of how spectators (users, in HCI terms), will experience the interaction and how they will react to it (which, many times in the arts is measurable and generalizable only to a small extent).

Iterating, creating different versions of the product that approach the intended product, is not only a proved methodology, but it also codifies something important: the assumption of the difficulty of new media creation and the need of progressive refinement.

Artists should also know that the context often plays a fundamental role in the characteristics of the interaction and try their prototypes in a space as close as possible as where the piece will be experienced.

User-centered design

User-centered design, where (some) users become part of the development team –because they are experts on their area of knowledge– is medullary to iterative design.

If with iterative design the team assumes that their work is perfectible and iterates creating many versions that get closer each time to their

[37] It is on purpose that we are not taking into account visual language related techniques (for example, those on Gestalt Theory), as its inclusion in an artistic or expressive environment feels a little too much.

objectives, user-centered design acknowledges that the team's objectives is to obtain the user needs.

In interactive art construction, involving the user –in this case, the spectator– is very much needed when the artist pretends a specific, measurable, reaction or interpretation.

Patterns

An important task of software engineering is the discovering and systematic use of design patterns, that is, a general reusable solution to a commonly occurring problem.

Also in interaction design, patterns are subject of research, and it is possible to find design pattern collections with the systematization of many interaction designs, solutions and schemes (see, for example, Tidwell's [125] or Fincher's [39]).

Once again, in the context of interactive art, it is not possible to propose a set of patterns that *should* be used. What we say is that some artists may find useful to know that the audience will interpret some interaction designs in a consistent way, or that in order to obtain some kind of reaction from the audience it is possible to apply some of these interaction patterns.

Psychophysics

One of the disciplines that are embraced by HCI, psychophysics –the branch of psychology concerned with the effect of physical processes (as intensity of stimulation) on the mental processes of an organism [83]– in particular when it studies how humans perceive art and how the sensory percept is endowed with aesthetics by the human brain has continued to fascinate psychologists and artists alike [131]. The artists are, then, confronted to the possibility of taking into account psychophysical data as part of their art creation processes.

Although there is some consensus on the fruitful convergence of new media art and HCI, as we have seen, the latter's goal of designing the best tools for a pre-conceived purpose is hardly fitting in art objectives.

There are, however, some gray areas where part (or the whole) of the artistic act is the development of a tool, being perhaps the most relevant example the construction of musical instruments.

Direct manipulation

Direct manipulation is an interaction style that stands for interactive systems with continuous representation of the domain of interest, with rapid, reversible, incremental actions and continuous feedback. Direct manipulation allows users to feel that they are operating directly with the objects presented to them with a direct representation of the domain of interest [93].

A direct manipulation system proposes a world metaphor [126], where the task domain is explicitly represented and the interaction language is implicitly coded in this metaphor. The orders users give to the system and their responses are implicitly coded in this metaphorical world[38].

The alternative is to offer a conversation metaphor, where the interaction language becomes explicit, and the task domain is mediated by descriptions in the conversation's language (e.g. a command line interface). Thanks to its verbal abstraction of the users' domain of interest, a conversation metaphor can provide a more powerful language of expression to its users, at the cost of a steeper learning curve.

When designing a direct manipulation based interaction, the representation of the task domain (the world) is a most important aspect of the systems appeal and usability. When users are not familiar (perhaps because it is a novel system) with the metaphorical representation, the design of this virtual world is key to the system's success.

Because of the difficulties of providing a well-designed world (both in terms of the objects presented and their interactive behaviors), some domains –like live video editing– have not been benefited by the multiple advantages of direct manipulation [31].

Direct manipulation-based systems can leverage the users' engagement and dexterity, making it a very valuable technique for designing, for example, interpretative instruments; a new media art area that we will now discuss.

Digital Lutherie

> *we risk having the whole field of interactive expression become an historical curiosity, a bizarre parallel to the true pulse of cultural growth. It needs all the effort and imagination that we can muster to assure that new controllers and interactive instruments indeed become the inevitable continuation of musical expression that we all take for granted.*
>
> Tod Machover [72]

---

[38] Traditional art practices, such as painting or sculpting, do show the mentioned characteristics of direct manipulation systems. This is because the interaction does not happens in a world metaphor but in the world itself, which shows something interesting: many times, the implementation of a direct manipulation system is *only* a process of virtualization.

One of the art fields more massively influenced by the digital is music, where new media poses new questions about music composition and performance.

Musical instruments, the tools for musical performance, are used to play and to produce music, transforming the actions of one or more performers into sound [58].

In addition, a luthier, as WordNet [102] defines it, is "a craftsman who makes stringed instruments (as lutes, guitars or violins)". However, this word has been used by extension to depict all kind of instrument makers, including those who create the more recent digital ones.

The lutherie –the work of luthiers– presents some of the classic assumptions about the relationship between art and technology: the latter enables the creation of tools to be used by artists in their art-making processes.

This is clear with musical instruments; traditionally, a musical instrument is created by a luthier and –on a posterior and almost unrelated step– is played by a musician.

This kind of relationship occurs because of the assumption of some implicit consensus between the luthier, the musician, and their social context.

First, they implicitly agree on how a particular instrument (like a violin) *should* sound. This also immediately gives us a way of measuring how good an instrument or a luthier is, for we can roughly measure how close an instrument is to its ideal.

They also agree on how a specific instrument is to be played, what kind of controllers, actuators, etc. it should have, and, also, its physical characteristics: weight, shape, etc.

Finally, they too agree on the social role that the instrument will play: how it is going to be played, its social and physical contexts and how the performance will be perceived by the public[39].

Although the study of traditional musical instruments exceeds the scope of this work, it is remarkable that all this assumptions are perfectly reasonable, taking into account that the luthiers, the musicians, and the public share an artistic language, from which the artwork emerges.

The field of digital lutherie –that is, the creation of digital musical instruments– on the other hand, starts from a much less solid ground.

[39] It is clear that with music, as with the rest of the artistic areas, there is no golden rule, nor *way* it should be done. In spite of this, what we mention here is correct for the popular music, that is, the vast majority of musical production.

In effect, if we see the creation of digital musical instruments as belonging to new media art, it can be argued that it has no predefined artistic language, and therefore, some of the assumptions present in traditional lutherie cannot be taken.

But perhaps more important is that there is no evident preconception on how a digital instrument should be played, nor how it should sound.

The artistic –semiotic– problem that artists face here is harder than what traditional music presents.

In effect, they are not only concerned about the music to play (taking music as the tonal and chronological arrangement of sounds) but also about how it will sound (the timbral aspects that are, to some extent, overseen in traditional music, or, more accurately, constricted to a set of pre-defined timbres, corresponding to the set of musical instruments and their nuances), and about how the instruments will be played and about how they will be presented to the audience.

Also –although this is not something that appears with new media art but, instead, something that has become much more common with it– the multimedia combinations of music and other stimuli, mainly the visual performances of live cinema.

The first and fundamental question that is posed here is if, and to what extent, the creation of instruments is part of the artistic endeavor or if it should be drown down to a previous and separate stage.

This question also applies to new media art in general: where (if at any place) one should draw the line that separates artistic creation to pure technology? Why, and how, one should try to find new tools, new media, new ways of communication, expression and art consumption?

The second question that we have been discussing in this chapter, namely, which roles can HCI play in new media art production, is especially relevant in digital lutherie. Being musical instruments *tools* for musical performance, it is reasonable to think that their design can be helped by HCI techniques and procedures.

Moreover, as the conception and design of new musical interfaces is a multidisciplinary area that tightly relates technology and artistic creation [57], one can suspect that HCI can play a role similar to the one that music theory does play.

When thinking about a new media art piece, there are no given boundaries: what the artist may be creating is, or can be, a performance of interaction between the artist, the technology and the audience. This interaction is multi-directional; if it is the intention of the artist to control it or to predict it, again, theory and knowledge of interaction can play their part.

Composed instruments

Digital lutherie explicitly blurs the divide between the creation of an art tool –the musical instrument– and the artistic production itself. According to Miller Puckette, "*the design of the software cannot help but affect what computer music will sound like*" an asseveration that we believe can be extended to all the aspects of digital lutherie.

For dealing with this, Bahn and Trueman introduced the concept of "composed instruments", which was later developed by IRCAM's Schenn and Battier in their 2002 paper:

> *A metaphor which is easily employed for a wide range of artistic performances with computers is that of the musical instrument. The term of the composed instrument underlines the fact that computer systems used in musical performance carry as much the notion of an instrument as that of a score, in the sense of determining various aspects of a musical work.*
>
> Norbert Schenn and Marc Battier [117]

We find here the assumption that instrument creation becomes part of the musical creation and therefore, new media musicians *compose instruments* as an integral part of musical composition.

There are two views on instrument composition that operate in orthogonal ways. Jordà proposes the following axioms [59]:

1. New music tends to be the result of new techniques, which can be compositional, instrumental or both.
2. New instruments will be able to survive and succeed as long as they are really innovative; i.e. they have something new to bring to music, and not merely because they are based on novel, previously unavailable technologies.

Under the idea that new music emerges as a consequence of new techniques, composers wanting to create new forms of music could start by creating a new instrument that, on a later stage, will enhance (or allow) their creative process.

The second approach conceives instrument composition as an optional (albeit new) step in the traditional work of a music composer. Under this conceptualization, when a composer is creating new music, he can, instead of *only* choosing the instrument that will play a certain part, create a new one in function of the desired sounds.

This is clearly orthogonal with the first view: in this case artists create the instruments that allow them to play their new music, while in the first one they create instruments to allow them to *discover* new music.

It should be clear that no approach should be, a priori, favored, as its effectiveness, in the end, could only be evaluated in function of the music that is created (or the visual aspects if those were more important), and the one fitting better will depend mostly of the compositional procedures and idea of each particular musician.

In spite of this, knowledge of interaction design should help us to face instrument composition and evaluation, especially if our focus shifts from music to interaction. Not only by using HCI techniques, but also as a source of inspiration for new interaction schemes.

HCI-based musical instrument evaluation and design

It is possible to classify live performance of music as a highly specialized field of HCI, dealing with very specific topics, such as simultaneous multi-parametric control, timing and rhythm. Therefore, it is possible to apply results from classical HCI in the evaluation of the proposed interaction [133].

One of the most classical techniques in HCI evaluation consists on the measurement of user success on the execution of very representative tasks.

With this in mind, Orio, Schnell and Wanderley propose the following set of musical tasks for interaction evaluation:

- Production of isolated tones, from simple triggering to varying characteristics of pitch, loudness, and timbre.
- Basic musical gestures: glissandi, trills, grace notes, and so on.
- Simple scales and arpeggios at different speed, range, and articulation.
- Phrases with different contours, from monotonic to random.
- Continuous feature modulation (for example timbre, amplitude or pitch) both for a given note and inside a phrase.
- Simple rhythms at different speeds combining tones or pre-recorded material.
- Synchronization of musical processes.

These tasks can be used to measure how a player performs with a given instrument.

Another common practice for evaluation in HCI consists in the utilization of quantitative methods that allow to measure a dimension of a given interaction gesture.

We will now discuss some of these quantitative methods.

Fitts' law. In his 1954 paper, Paul Fitts proposed a formal relationship to describe the relation between performance and accuracy for the simple task of target pointing. One formulation of the law is [74]

$$T = a + b \log (A/W + 1)$$

where T is the lapse needed by the user to perform the task of pointing to a target of width W from a distance A. Constants a and b are empirically determined.

The task of pointing is very common in graphical user interfaces (for example when using a computer mouse to press a button drawn in the screen), and Fitts' law allows us to, for example, predict how big a button should be if we want to bound the time users will need to press it with their cursor starting in any place of the screen.

GOMS. GOMS, an acronym for Goals, Operators, Methods, and Selection rules, is another quantitative method of proposed by Card, Moran and Newell in their 1983 book *The Psychology of Human Computer Interaction* [21].

The family of GOMS-based methods, such as CPM-GOMS and NGOMSL, models the interaction as a string of atomic physical, cognitive or perceptual actions (usually not taking into account some interaction characteristics, like tiredness, alternative cognitive paths, mistakes, etc.).

One example of GOMS utilization consists on considering the following atomic elements for computer usage, and their estimated duration:

- Keying (K): 0.2s – Time needed for typing a key.
- Pointing (P): 1.1s – Time needed for pointing with the mouse to one location on the screen-
- Homing (H): 0.4s – Time needed for switching the keyboard for the mouse and vice-versa.
- Mentally preparing (M) 1.35s – Time needed by the user for preparing for the next cognitive step.
- Responding (R): computer-dependant. – Time the user waits for the computer to respond.

GOMS, then, proposes a set of rules for creating interaction strings, like inserting a M before all not-anticipated operators, or before the sets of Ks for each cognitive unit (such as words), etc.

Once we have the strings for our to-be-measured interaction scheme, GOMS predicts how long the interaction task will take for an average user.

Steering Law: A generalization of Fitts' Law proposed by Accot and Zhai [2] that predicts the time needed by the user for a more complex task that in the case of Fitts: to navigate through a two-dimensional tunnel (thus, steering to follow the path).

The steering law for a generic curved path can represented by the following equation:

$$T = a + b \int_{C} ds/W(s)$$

where T is the time to move through a curved path C, with variable width W(s) and *a* and *b* are constants.

These predictions, and any other that may appear under the same principle (namely, a simple quantitative relationships between task constraint and movement speed), can be used in musical instrument creation and evaluation. As an example, a GOMS-like technique can be used with the set of tasks proposed by Orio et al, and –especially in the case of computer-based instruments– both Fitts' and Steering laws can be used to test a given gesture.

Other digital luthiers have attempted to use HCI techniques and results in instrument creation, so Fernandes and Holmes [38], showed the redesign of musical hardware in function of heuristic-based user performance testing, and Vertegaal compared several input devices in a timbre navigation task by comparing users' movement time and errors while navigating a four-dimensional timbre space [129].

Although we state that pure HCI can help musical instruments design, music performance clearly constitutes a very-specialized and charismatic field, where much of the interaction is particularly hard to model, and its nuances and details are of a precision that is not common on HCI.

While it is true that many specialized fields of interaction research need a highly accurate model an formalization, like all life-critical fields, it is also true that very few need the freedom that musical performers require, and those fields that do need finely crafted interactions (for example, medical surgery), often rely on very simple tools and let the user make all the cognitive effort (much like in traditional musical performance).

In the subtleties of musical performance there are non-obvious performer gestures that often influence the final sound produced [132] and that fall apart of current HCI gesture taxonomies, reinforcing the idiosyncratic approach commonly associated with creative and expressive designs and instrument composition.

With this caveat on mind, we re-state that not only interaction design may aid in instruments' evaluation but also can play a defining role in their creation.

If we accept Jordà's second axiom on digital lutherie we are forced to innovate in musical instrument design, and one possible field of innovation is the very act of interaction.

In effect, Chadabe[40] states that the traditional mapping, that is, the description of the way a performer's controls are connected to sound variables is only really useful for instruments modeled after traditional ones. In innovative, interactive instruments, mapping appears to be less useful, thus requiring new interaction techniques [22].

When composing a new instrument, artists face new characteristics to explore, some of them being level-of-abstraction, timbre, chance, and gestures.

Partially thanks to the independence between controllers and sound generators that electronic instruments provide, the instrument designer can incorporate new amounts of chance into the music production, shifting part of the cognitive process from the performer to the instrument[41] and, therefore, increasing the abstraction level at where the performer is operating.

This can happen on a global basis (the musical description that the performers affects is increased on every variable) or locally, affecting only some of the musical variables.

This increase in abstraction can be mapped onto a taxonomic line ranging from fully deterministic, where once the musical characteristics and limitations of the instrument are apprehended, the (virtuous) performer has complete control of the instrument's output, to completely nondeterministic, where no control can be exercised by the (now virtually nonexistent) performer.

Is when one traverses the taxonomic line towards the nondeterministic end where the traditional mapping ceases to be applicable; it is also there where HCI techniques of interaction design can help the instrument composer.

---

[40] Joel Chadabe, composer, author, is an internationally recognized pioneer in the development of interactive music systems. He has concertized widely since 1969, in all major venues. As president of Intelligent Music from 1983-1994, he was responsible for the development and publication of a wide range of innovative and historically important software and hardware. He was keynote speaker at the NIME 2002 and is author of the book 'Electric Sound: The Past and Promise of Electronic Music'. He is currently Professor Emeritus at State University of New York at Albany; Director of the Computer Music Studio at Manhattan School of Music; Visiting Faculty at New York University; and Founder and President of Electronic Music Foundation.

[41] Interestingly, it can be argued that the cognitive shift is towards the instrument's composer, and not to the instrument itself, therefore introducing the concept of asynchronous (and perhaps unknowingly) *joint* composition. This is also implied in the concept of *composed instruments.*

To work in non-deterministic instrument design, Chadabe proposes the fly-by-wire metaphor. In aviation, 'fly-by-wire' describes a system in which a pilot tells a computer what the airplane should do and the computer, to some exempt, autonomously flies the plane.

This metaphor can be applied to delegate aspects of the performance to the instrument (a trivial example would be the automatic accompanying systems that many electronic keyboards have).

Another area of HCI expertise is the explicit management of multi-modal interaction. It is clear those different axes do exist on traditional musical performance (such as tone and intensity), but are traditionally presented orthogonally.

Interaction designers have put a considerable amount of effort on multimodal interaction, for example Tanaka and Knapp [123] concentrated on the fusion of many inputs from different domains towards a given task.

They propose three design goals that allow for successful multimodal musical interfaces:

- All of the component modes are intuitive interfaces.
- The multimode context leverages the richness of each interface to expand the articulative range of the other.
- The two interfaces are independent and yet exhibit bi-directional complementarity.

They also introduce the notion of “bidirectional complementarity”, as the interdependent relationship between two (or more) sensing systems, and establish the richness of interaction required and afforded by music.

According to them, artists can create instruments that operate on a multimodal and synergic way (opposed to the traditional orthogonality), while retaining the intuitiveness and expressiveness of traditional musical interfaces.

Low cost digital lutherie

When the different approaches on new media art or, more specifically, digital lutherie are discussed, it is implicitly assumed that artists have the *means to the media*. No matter what the medium is or how it fits into the art-production scene (a raw medium, a tool, etc.), they must be able to have access to it in the very first place [67].

This obvious necessity is wildly overseen in the literature, which does not take it into account when talking about the democratization (of access to, or creation of, information or art) that technology implies.

We can realize that this is indeed a very serious problem by taking a look at the prices involved. For example, two of the most famous tools that attempt to help technological appropriation, like MAX and MSP[42] cost 395 USD[43], a price that cannot be afforded by a significant part of the artist population or the art institutions.

In a context of this kind (i.e. with virtually no budget except for some old hardware already available), we attempted to create the *Tecnocordio*[44], a mixed media, networking-enabled, gesturally controlled, musical instrument that we will now describe.

It is perhaps worth noticing that we had no previous experience with electronics, something we consider particularly important as it shows that other artists with no such background can use the type of techniques we will show[45].

The *Tecnocordio* was a mixed-media musical instrument created using inexpensive, obsolete and recycled technology. In the Hornbostel-Sachs [53] musical instrument taxonomy it would perhaps be a chordophone (a string instrument) although it also generated synthesized sound and real-time images.

One important aspect of mixed-media musical instruments is to recover the role of the visual appearance of the instrument, something often lacking in all-digital musical instruments.

The *Tecnocordio* was constructed with two main goals in mind: to test the expressive and functional capabilities of technological garbage, and to see if these capabilities were enough to create a musical instrument.

After deciding to only use recycled hardware, the first problem to be solved was the actual sound making. This was resolved by using stepper motors to strung guitar strings attached to a resonant wood frame.

Although in the whole design of the instrument, our main concern was cost, some computing power was to be needed. We then restricted ourselves to use only the hardware we already had in the lab (recycled with the techniques to be described in part 3), which also included one rather modest motherboard and processor (a 500 MHz Pentium III PC) and some much older mothers and processors (66 and 100 MHz 386).

Having its main source of computational power in the Pentium III, the *Tecnocordio* adopted the form of a client-server system, where the server

---

[42] Discussed in the next chapter.

[43] Price as of 2007-03-06, source: http://www.cycling74.com

[44] The *Tecnocordio* was created by Fabrizio Castro and this thesis' author for our degree on Computer Engineering at the Universidad de la República, in Uruguay.

[45] It wouldn't be fair to omit that we did have some help from the staff of the IIE, but we still believe that this help could be substituted with deeper internet searching.

interpreted the inputs (scripted or real-time) and sent orders to the clients (the older PCs), which, in turn, controlled the motors via their parallel port. This distributed approach was also desirable as it would allow us to create geographically disperse versions of the instrument.

We also wanted to allow as much input techniques as possible, to stress the HCI capabilities of the hardware. Therefore, in the *Tecnocordio* we implemented control (see Figure 7) via MIDI, basic visual gestures (using a cheap web-cam), real-time scripting, and off-line scripting (which we refer to as *scoring*).

The instrument itself consisted in a metallic structure (made also out of recycled PC cases), which held the motherboards, the strings and the motors (these last two in a resonant wood frame). Each motherboard controlled four motors, being then able to play four strings.

All the PCs run Linux (a stripped-down version of Slackware Linux that fit in the 20 Mb HDDs available at the lab) and had old network cards that allowed us to set up a 10-baseT LAN, linking them to the server.

The server ran our controlling software, written in Java, and allowed the instrument to be controlled in by Java scripting (using BeanShell[46]) and MIDI.

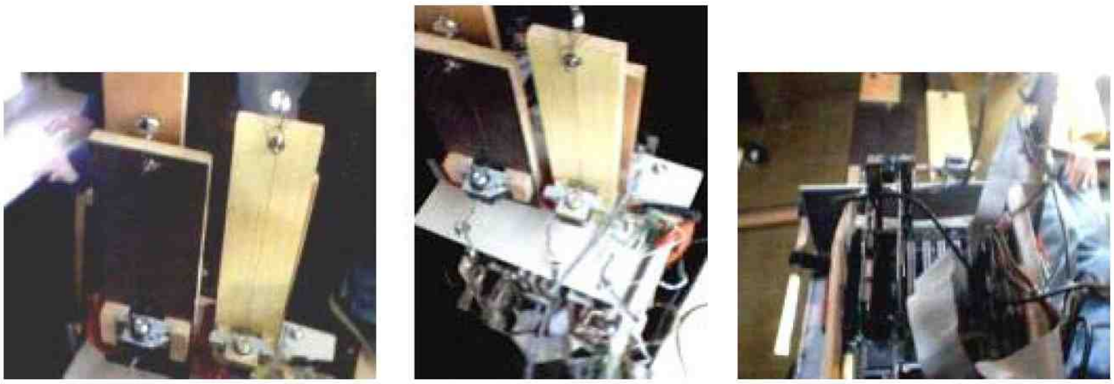

Figure 6 - The *Tecnocordio*. Left: two strings and two stepper motors. Middle: more strings and the base with the motherboards. Right: the motherboards.

MIDI control of the *Tecnocordio* was solved using the server's MIDI-in port. The server software translated some of the MIDI commands (only note-on and note-off were implemented) to the corresponding[47] commands to be sent to a client.

[46] http://beanshell.org

[47] For this to occur, a mapping (IP, motor#) to MIDI note had to be previously set using the configuration files or by a script.

Only note on and note off were supported and all note-on were played at maximum velocity (MIDI argot for intensity).

If *scoring* were to be used for controlling, the user had access to the motors by using the mapping but also in direct form, being then able to write commands of the form playNote(A4), and playNote(IP, motor#).

It is worth noting that our implementation corresponds to a more general software architecture, where different inputs can be arbitrarily mapped to different outputs. In that framework, linking MIDI-in to the *Tecnocordio* was only a matter of configuration.

It is also to note that although we fully implemented the architecture for the *Tecnocordio*, MIDI, scripting, software synthesizer, etc., the *Tecnocordio* itself never surpassed the prototype stage. It is probably its lack of novelty, in musical terms, the main reason for it not surviving.

Low-cost crafting techniques

Although the creation of the *Tecnocordio* and its architecture conferred great joy to us, we believe that the main lesson to be extracted from this work is that one can create a physically-based musical instrument with a little budget, and with just the most basic knowledge of electronics.

As we stated before, our objective was to create a musical instrument by using only the hardware already available in our lab at that time, which we suppose was a fairly standard third-world electrical engineering lab. It is our hope that this way of approaching the problem can be used by others to attempt more sophisticated endeavors.

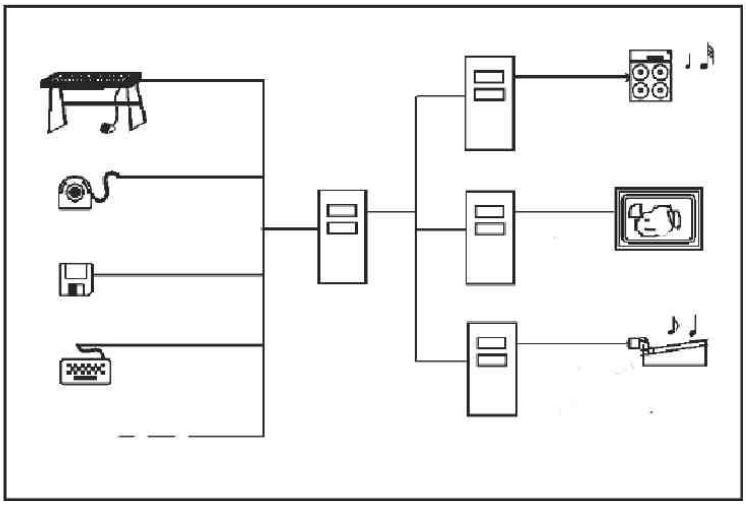

Figure 7 - The *Teconcordio*'s I/O; on the left the inputs, MIDI, gestures, scoring, scripting; on the right the outputs: synthesized sound, images, physically produced sound

Knowing that the *Tecnocordio* was to be a string instrument, the first problem we faced was to play the strings. We decided to stick to the simplest setup: a motor used to hit a guitar string, which would then vibrate in its fundamental note (see Figure 8).

We used stepper motors recycled from old 5"¼ diskette drives that fell to no use. As we had no programmable hardware (e.g. a PIC), we created our own hardware drivers for controlling them, by recycling both the needed chips and the PCB.

These motors have four coils that can be excited separately. By doing it in order we can get the motor to spin. A simple circuit for doing so is depicted in Figure 9, but for implementing it, we needed some XOR gates and some way to obtain the 12V[48].

The ULN2003 chips [122] have seven Darlington arrays that we can use for leveraging the voltage and the 7486 [124] family of chips have four XOR gates for controlling the logic.

What makes this particularly apt is that both of them can be found in many outdated hardware, and we happened to have them in the boards of the same diskette drives. To use them we de-soldered them using a paint-removing gun, and then re-assembled them.

The next step was to create a circuit with these chips. Our choice was to turn to the leftovers of our de-soldering and to cut and paste the PCB.

Figure 8 - A string and its stepper motor

As is shown in Figure 10, it is possible to craft connectors and circuit boards by cutting and soldering PCB from old boards.

By de-soldering chips, cutting PCB and recycling motors we had our basic modular piece of the *Tecnocordio*.

As the steppers were to be controlled by only one performer, we used our only *powerful* PC to act as the CPU of a distributed instrument, where the IP nodes had each four musical notes.

[48] The parallel port outputs 5V.

Being able to (at least partially) recycle and reuse obsolete technology for specific (and often not very processing power demanding) tasks can be a strong way of fighting against both the digital divide and the so-called e-waste.

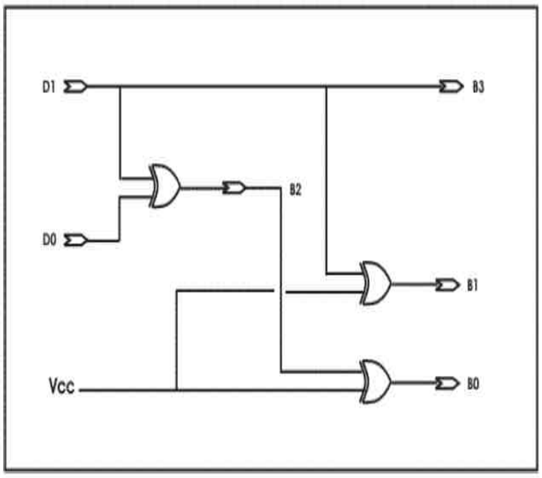

Figure 9 - A very simple circuitry for controlling steppers.

However, one should not forget that there are many problems that do require state of the art technology. We believe that old, recycled technology and new one can coexist and collaborate via networking. For our instrument we set up a TCP/IP network over coaxial cable that allowed us to have obsolete and new technologies interact seamlessly (while at the same time letting us distribute physically the *Tecnocordio*: the different strings and components could be arbitrarily distant).

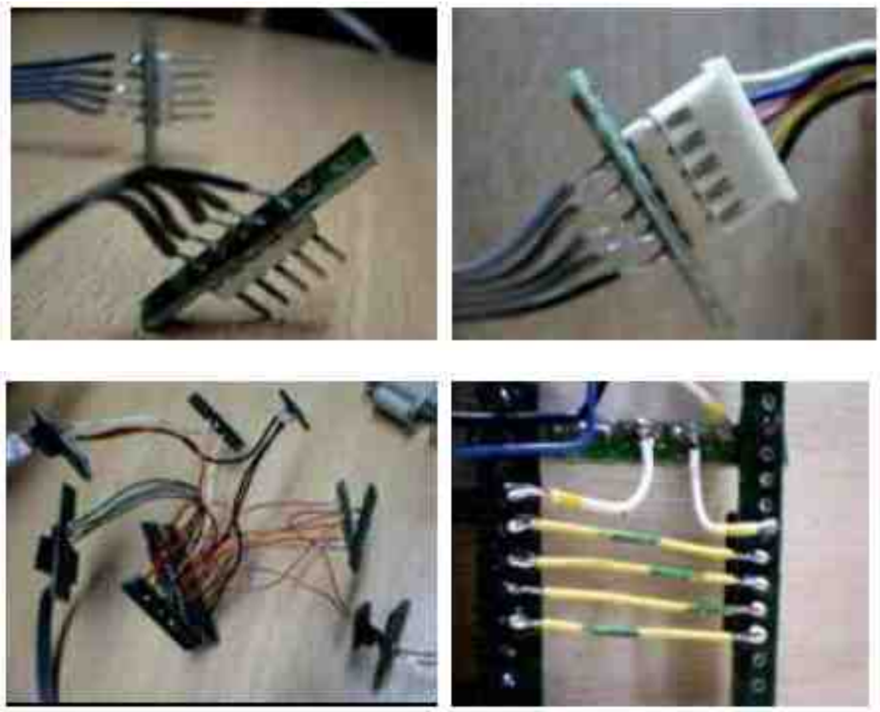

Figure 10 - Crafting PCB

Also, if we were to consider the *Tecnocordio* as a real musical instrument and not an exercise or a proof-of-concept, further –especially musical– evaluation would be indispensable.

However, we do hope that this attitude towards technology can not only help to bridge the digital divide by providing new means of creation, but also be of help in the creation of new artistic languages that profit from the direct manipulation of low-level technology as a medium, and see themselves not as a copy of the modern digital language, but as an authentic, location-specific alternative.

The art of interactivity

> *Since 1969, I have been trying to raise interactivity to the level of an art form as opposed to making art work that happened to be interactive.*
>
> Myron Krueger [84]

Myron Krueger's[49] quote raises a crucial issue: the aesthetics of interaction.

If interactivity is as important as being considered "the defining feature of new media installation" [112] it cannot be relegated to play an accessory role but, instead, it must be considered as a central part of the aesthetic experience.

In a clever re-statement of Marshall McLuhan's famous dictum "the medium is the message", David Rokeby poses that "interface is content" [113], yet, we believe that *interaction is content* would suit best.

In effect, the very notion of interactivity questions not only the aesthetic characteristics of both art objects and the artistic experience but also the artists' relation with the oeuvre and the public. In the words of Martin Rieser[50]: "they [the art objects] can only become truly interactive when

[49] Myron Krueger (b. 1942) is an American computer artist who developed early interactive works. He considered to be one of the first generation virtual reality and augmented reality researchers. He coined the concept of "artificial reality" and predicted what is now known as virtual reality (and its appliances). His most famous work, *Videoplace*, has been exhibited widely in both art and science contexts and was the featured exhibit at SIGCHI (Computer-Human Interaction Conference) in 1985 and 1989, and at the 1999 Ars Electronica Festival.

[50] Martin Rieser (b. 1951) is a British researcher and artist. He has exhibited and presented papers widely and has curated various exhibitions including 'Electronic Print', the first international exhibition of its kind. He is co-editor of new Screen Media; Cinema/Art/Narrative and currently works at Bath Spa University College at Bath School of Art and Design as Professor in Digital Arts. He set up one of the first post-

authors attempt to transcend the established syntax of earlier forms and the platitudes of multimedia and invent a coherent artistic language for interaction" [112].

By stating the need of an *artistic language* of interaction, we not only acknowledge the early tendency of new media art to require spectators to actively *complete* the artwork with their actions [54], but also the need of an "expanded definition of interactivity" [26] that effectively allows for all possible instantiations of the interaction in the artistic search of aesthetic meaning.

graduate courses in the UK in Digital Art and Imaging at the City of London Polytechnic in 1980-85.

3 PROGRAMMING ART

When Communications of the ACM began publication in 1959, the members of ACM'S Editorial Board made the following remark as they described the purposes of ACM'S periodicals 2: "If computer programming is to become an important part of computer research and development, a transition of programming from an art to a disciplined science must be effected." [...] Meanwhile we have actually succeeded in making our discipline a science, and in a remarkably simple way: merely by deciding to call it "computer science."

Donald E. Knuth's Turing Award acceptance speech [64]

Computer programming plays a defining role in new media art. Manovich's third principle[51] states that new media art is capable of automation; this implies the delegation of some –sometimes all– of the cognitive processes to automatic procedures. This delegation can be achieved by computer programming.

However, what do we understand by programming a computer[52]? What are the differences between programming a computer and using one?

Princeton's *WordNet* defines computer programming as the creation of a sequence of instructions to enable the computer to do something [102]. Although all definitions we have seen are similar, many also name the sequence of actions "source code".

This name is misleading, as it suggests traditional, textual programming languages that are used on a stage *previous* to the execution of their products (that is, products built with them).

This is true even for interpreted languages: there is a stage of source code writing and one or more subsequent stages that make the computer run that code (which may or may not interact with its users).

However, if we reflect on the definition we agreed on, to program is to *specify a set of actions that a computer must perform*, it is easy to see that any interaction language between users and computers is a programming language: it specifies actions to be executed (often immediately) by the computer.

To use a computer is, always, to program it.

[51] For a discussion on new media art and Manovich's principles, please refer to chapter 1, *New media art*.

[52] A computer, according to the Merriam-Webster dictionary is "a programmable usually electronic device that can store, retrieve, and process data". Hence, we do not need to define what a computer is, as long as it is a programmable thing.

In effect, when people use a computer, they are –by means of the interaction language the computer interface proposes– specifying what actions it must perform; users give orders to computers when they double-click on an icon or when they write down the actions in a programming language.

We can also find direct manipulation programming languages, such as those used for designing interfaces. Even though they clearly conform a programming technique, they are comprised of the interaction gestures of end-user interaction, and therefore constitute a class between textual programming and end-user interaction.

The difference between what the literature traditionally catalogues as programming or as using a computer is not qualitative but a quantitative one and refers to the *freedom* the user perceives.

Freedom is illusory. Users, even if they consider themselves programmers, are always constrained by the expressive power of the interaction language, the computational power of the underlying hardware and their ability to express their ideas in that language.

A similar discussion can be carried on comparing musical instruments and devices of music reproduction such as CD players. They are basically, qualitatively, the same type of machines (sound production ones), and the difference between them is the perceived freedom, that is, the perceived aptitude that the machine has to adapt itself to its users needs or, in other words, the versatility of the interaction language –in a given context– the instrument proposes.

There is a strong relationship between this instrumental freedom and the abstraction level of the production of the interaction phrases. Highly abstracted gestures can produce information-dense outputs: such gestures can be to press the "play" button on a CD player or to double-click on an icon to run a computer program.

On the other side of the spectrum we would have very basic gestures (stringing a violin string, or writing a line of code in C), whose combination would allow users to shape the desired output.

But it is not as easy as tracing a continuum between [more freedom, less abstraction] to [less freedom, more abstraction], as freedom also (if not mainly) relates to the users' ability to obtain the desired performance from the tool.

Direct machine language programming allows, theoretically, for the widest range of results but requires users to have an important amount of knowledge, much more than with other higher-level interaction languages (and also, probably requiring more time than users wants or can assign to the task).

As John Maeda[53] puts it:

> *Users of tools are much more prevalent than makers of tools. This imbalance has traditionally been rooted in the vast difference in skill levels required for using a tool compared to making a tool: To use a tool on the computer, you need do little more than point and click; to create a tool, you must understand the arcane art of computer programming.*
>
> John Maeda, Creative code [77]

We can also see that there is a similitude between music re-mixing or DJing and commercial software scripting, for they both use pre-created material as creative tools.

It can be reasonably argued that every new media production is built on top of other creations, tools, or products and, therefore, can be seen as a kind of remix. For example, a program written in the *Processing* programming language, when executed is translated onto a *Java* program that is compiled –translated again– to *bytecode,* the machine language of the Java Virtual Machine, which is interpreted by it and translated to orders to the operating system. The operating system will itself translate once again the orders to hardware calls (more orders), which, at the lowest level, will be carried on by one or more processors, that implement only the operations the machine actually knows how to perform.

But if to use a computer is not fundamentally different from programming it, what are the differences between tools and products and what are the differences between users and programmers?

This is indeed a fair question, and again, the difference is perceptual: it is a difference of interpretation, or of contextual situation and not an

[53] John Maeda is the president of the Rhode Island School of Design, a world-renowned graphic designer, artist, and computer scientist, who has pioneered the use of the computer for people of all ages and skills to create art.

Maeda's early work helped redefine the use of electronic media as a tool for expression by combining skilled computer programming with traditional artistic concerns.

He is the recipient of the highest career honors for design in the USA (2001, National Design Award), Japan (2002, Mainichi Design Prize), and Germany (2005, Raymond Loewy Foundation Prize). In May of 2003, he received an honorary doctorate of fine arts from the Maryland Institute College of Art. He received both his BS and MS degrees from MIT, and earned his PhD in design from Tsukuba University Institute of Art and Design in Japan.

inherent one[54]. The difference lies on the relationship with the technological pieces that people construct.

Media appropriation –that we have discussed before and lets technology users become creators instead of pure[55] consumers– can be seen as turning final products into tools, and tools into media, to re-articulate and re-mix products, to recover their expressive, functional and semiotic capabilities and create new ones.

As Knuth's quote shows, it has been discussed whether computer programming consists an art form itself, something still debated nowadays [71] [55]. Interestingly, this discussion only arises in technological or scientific environments, and it happens so because –from art's point of view– computer programming, appropriated, has blended into art production, rendering futile the discussion of if, or when, it constitutes art.

The appropriation of extremely refined technological activities, such as programming, has been steadily happening. Computer programming has changed from a *lingua franca* (a language that is used for more people than its native speakers but usually in a not very proficient way) to the *vernacular language* of new media art.

We then can find many examples of highly refined art that show a deep understanding of computer programming, together with the hybridism of new media artists.

But, as happens with many, if not all, have the gestures involved in artistic expression, computer technology is very demanding of specific knowledge from the artist.

In the same way that traditional musicians do need training on the interaction with their instruments, in order to be able to code the intended performance into the instrument's language, new media artists need to be able to code into –and hopefully even redefine– the language of new media.

Artistic languages can be thought as defined by the act of coding: the scope of use and expression of a tool or medium is not inherently, or a priori, bounded.

---

[54] Traditional computer literature has always assumed a difference between computer programmers and users, where the former create tools (software) for the latter. This difference has been blurred with the assumption of the software's need to offer scripting or macro-sentence creation possibilities.

[55] Pure consumer is an blunt exaggeration, as there is a certain amount of creative work in using any cultural product.

Despite the existence of "happy coding mistakes", there is a need to be fluent in both the arts and the information technologies [98], which forces artists to be able to cope with the skills that computer programming requires.

As a response to this necessity, there are several attempts to both ease the associated learning curve and to maximize the expressive power of the language. These attempts range from low-level APIs and programmable modular hardware, like Arduino Software's *Arduino*, to real-time capable visual languages like Puckette's *Pure Data*.

Aesthetic computing

However, it does not suffice to catalog artistic or expression oriented computer languages only as simplifiers that aim to ease computer-programming learning.

New media appropriation implies the recontextualization of the scientific and technological aspects of new media, adding an aesthetic dimension to them.

This recontextualization poses new questions on both the artistic and technological realms, offering a rich field for hybridization and cross-fertilization called aesthetic computing.

Aesthetic computing, a term coined by Paul Fishwick[56], can be defined as the application of art theory and practice to computing [41].

Even though aesthetics is present in every human activity[57], the cultural patterns that shape our taste differ in disparate activities. In formal environments, such as engineering or mathematics, aesthetics is usually synonymous with optimality [14] or effectiveness [28], and therefore, the addition of the aesthetic perspective of the arts constitutes an "emancipatory praxis where unexpected things can happen" and where software can go "beyond the designed purposefulness" [14].

---

[56] Paul A. Fishwick is Professor of Computer and Information Science and Engineering at the University of Florida. He received the BS in Mathematics from the Pennsylvania State University, MS in Applied Science from the College of William and Mary, and PhD in Computer and Information Science from the University of Pennsylvania in 1986. His research interests are based in modeling, simulation and computer arts. He is a Fellow of the Society for Computer Simulation (SCS), and a Senior Member of the IEEE. He has chaired workshops and conferences in the area of computer simulation, served on numerous journal editorial boards and published numerous books and papers.

[57] We would like to quote Albert Einstein: "The most beautiful thing we can experience is the mysterious. It is the source of all true art and all science", and John Keats "I never can feel certain of any truth but from a clear perception of its Beauty".

According to the model of aesthetic computing[58], this bidirectional appropriation[59], which Fishwick calls "the art-computer integration", occurs in three levels: cultural, implementation, and representation [42], as follows.

Cultural level

Cultural level is where computer programming is affected by its artistic -–aesthetic– context and refers to the necessary and unavoidable changes in technology that appear as a consequence of its artistic appropriation.

As we have shown, the utilization of technology as a raw medium of artistic production not only allows for its subversion, but also for the creation of artistic languages that do not have to comply with the dominant trends [67].

By addressing the existence of the cultural level in the model of aesthetic computing, we are making explicit the interaction and cross-fertilization between art and technology, with the welcomed side-effect of freeing new cultural developments from the constraints of pre-existent software [66].

Implementation level

The implementation level consists of the *behavioral coupling* between specific coding gestures or artifacts and artistic or design equivalents. Each computing artifact, if executed, renders an artistic consequence.

This trivial fact allows for the creation of *coding gestures* where specific code fragments are used and re-used to create specific artistic artifacts. While sometimes the over-use of these fragments tend to some homogenization of the production, the gestural approach allows for the inclusion of classical HCI and art techniques such as early prototyping and drafting.

Its over-use also questions whether true appropriation occurs or if the user is more-or-less blindly reproducing gestures crafted by others. This behavioral coupling would allow for an intermediate state where artists

---

[58] As occurred with Lev Manovich's new media principles, in this work we present our conceptions of aesthetic computing, which –although based on them– do not necessarily agree with all Fishwick's (or others') work. We encourage the reader to read Fishwick's (and Manovich's) original works.

[59] Aesthetic computing, as we can see from its name, "combines two aspects of reality in an asymmetric way. Its syntax indicates that 'computing' is the governing aspect" [90]. Even though it is a misnomer to some exempt, the asymmetry reflects a broad an active field of work that does stress the computing side. This is why we confine our mentioning of aesthetic computing to this chapter and did not include it in the first, whose discussion we pretended to be more general.

can remix pre-existent portions of code but are not able to create new code. However, we are all in different levels of this state, as one does not program one's own operating system or creates one's own computer hardware

It is also interesting to note the relationship between code and design, in Fishwick's words:

> *There is a tight coupling between code and design: for any piece of code, there is a commensurate, behaviorally generated, visual design. This creates a strong bond to where the visual pattern becomes partially synonymous with the concept of conditional iteration associated with the computing artifact.*
>
> Paul A. Fishwick [42]

Representation level

While the implementation level refers to the behavioral coupling between coding and design, the representation level models the *structural coupling* between them.

However, modeling the relationship between art and technology, or, better, design and code, as if they have always presented a significant degree of structural pairing would be too reductionist.

Indeed, for structural pairing to appear naturally, an established methodology and artistic language must preexist. It is possible to see many examples of artistic appropriation that subverts not only the intended use of a given technology but also finds new aesthetic values in preexistent technical production and designs.

But, as it happens with behavioral code gestures (specific code fragments used and re-used), it is also possible to identify structural tendencies that, within some design approaches, do show a very precise pairing.

Information visualization [42; 134], where the graphical representation of data acquires an aesthetic side that sometimes prevails over the utilitarian, communicational, aspects of it, is a fine example of representation pairing, where the design artifact do show the structural characteristics of both data and code.

The code used for data representation becomes an inextricable part of its visualization, of its aesthetic valuation.

Programming languages

> *As programming becomes easier and more accessible, the tools for expression are becoming more complex and difficult*

to use. Programming tools are increasingly oriented toward fill-in-the-blank approaches to the construction of code, making it easy to create programs but resulting in software with less originality and fewer differentiating features.

John Maeda, Creative code [77]

An obvious requisite for computer programming is to have a programming language to code with. Which languages are best suited for learning how to program is a problem we won't be tackling, as there is a significant amount of literature discussing it (see, for example, [62]). Instead, we want to engage in a discussion of the characteristics of the most important (in terms of impact and popularity) art-oriented programming languages.

Being particularly concerned with Maeda's quote shown above, we ask: is media appropriation hindered by learning-curve easing tools, are they a useful middle step towards full appropriation, or are they adequate tools that fulfill new media artists' needs? Is the structural pairing of aesthetic computing's representation level a homogenizing force?

As we have discussed before, it is clear that art in general, and media art in particular, is completely dependent of the use of tools, what changes with media appropriation is the nature of the relationship between artists and tools.

A new realm of art-creation processes appears when the invention of new tools becomes an integral part of it, as in the already mentioned[60] idea of composition of instruments.

But even in the most pre-eminent cases of skilled technology appropriators, it is impossible to have a fine knowledge of all the technology involved; for example, programmers might not know about the details of their computer's hardware, or, if they do, they might not know as much about solid-state quantum physics[61].

Visual languages

Visual programming languages construct their proposal on the assumption that human information processing is inherently analog and visual: people relate to the world in an inherently graphical way, use

[60] See *Composed instruments* in chapter 2, *Human-computer interaction*.

[61] This is clearly addressed by Roger Malina when he mentions that the "new Leonardos" may by interdisciplinary groups, while Maeda believes that new media artist should be "real hybrids", that is, to possess a fair amount of knowledge in all the involved areas.

imagery as a primary component of creative thought, and analog modalities of communication predate digital (verbal) ones [135].

These two assumptions have led language and interaction designers to propose languages with *analog, graphical representations*[62] of the software that is being written, thus placing a particular accent on the representation level, even before the product is constructed.

Two of the first visual programming languages were Ivan Sutherland's *Sketchpad* in 1963 (see Figure 11) [15] and Canfield Smith's *Pygmalion* in 1975 [70].

Sketchpad was a truly groundbreaking system –for which his creator was given a Turing award– and set the basics for what we now know as graphical user interfaces, computer-aided design (its XY-based interface is still in use today) and object-oriented programming, introducing both the notions of classes an instances.

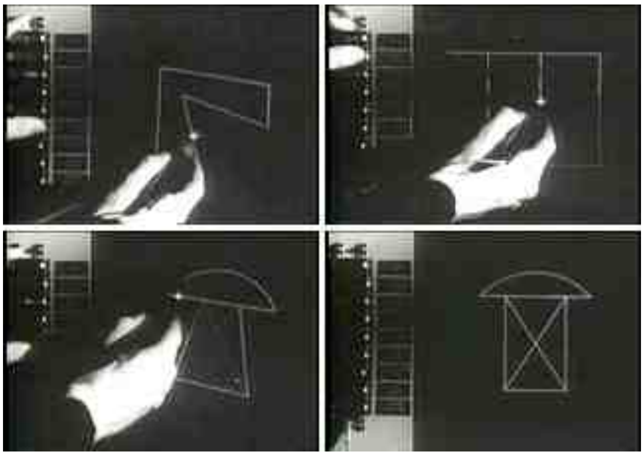

Figure 11 – Ivan Sutherland showcasing Sketchpad [61]

It can be said that *Sketchpad* was the first object-oriented programming project, that it had the first use of a toolbar, the first real-time graphics system, the first drawing program, the first graphical user interface, the first use of instances and the first use of "draggable" vector graphics.

[62] A commonly used definition of visual programming language is that it consists of "any system that allows the user to specify a program in a two (or more) dimensional fashion. Conventional textual languages are note considered two-dimensional since the compiler or interpreter processes it as a long, one-dimensional stream. This definition does not include any system that uses conventional (linear) programming languages to define pictures, thus eliminating most graphic editors, like Sketchpad." [88] We won't use this definition, nor any of its kind as it focuses on how the computer represents the program (which, in the end will usually go down to a linear stream), instead of focusing on how users conceive it, and how they interact with the development environment. Programming is a matter of human-computer interaction, rather than a matter of compiler design.

However, where *Sketchpad* stated the basics for graphically mediated human-computer interaction and object-oriented computer programming in 1963, *Pygmalion* did the same for visual programming, twelve years later[63] [70].

Canfield Smith's work, titled *Pygmalion: A Creative Programming Environment* was also very innovative: it introduced the notion of icons, drag-and-drop as a way of passing arguments to functions, and programming by example.

Although it is possible to identify other works in visual programming between *Sketchpad* and *Pygmalion* –such as Ivan Sutherland's brother's, Bert, dataflow language– *Sketchpad* and, especially, *Pygmalion* where the first two explicitly oriented towards creativity and artistic creation.

This apparently natural usage of visual programming languages to perform artistic activities couples with the characteristic that, for researchers such as Scrivener [119], Myers [88], or Burnett [18], visual programming languages can be easier to learn and to use (or even more fun to use!) than their entirely textual counterparts[64].

These characteristics are, of course, desirable not only in artistic or expressive environments, and visual representation of algorithmic and computational constructs has always been present in programming. As a matter of fact, "most of the analogue machines which were constructed prior to the development of digital computers were programmed in a pictorial fashion" [4].

Therefore, visual programming has been successful in several areas, in particular those areas where the graphical, explicit representation of the domain of interest is natural or, at least, possible.

Perhaps the most wildly known examples are graphical user interface editors (such as the form editors) popularized by Microsoft's Visual Basic. However, in other areas, like electronics (low level programming) and network design, visual programming is also very popular.

A taxonomy of visual programming languages is often constructed based on its visual purity. In function of it they are sub-classified as

[63] Interestingly enough, Canfield's advisor was Alan Kay, who –in turn– has had Sutherland as his doctoral advisor.

[64] This contraposition between textual and pictorial representation makes us remember Vílem Flusser's *Towards a Philosophy of Photography,* see *Augmented reality* in [43], where he presents the evolution from pictorial representation to textual representation first, and to technically produced (i.e. by means of an apparatus) pictorial representation later. More specifically, Flusser catalogs the technical image as *applied scientific text*, a concept that becomes interestingly explicit when there is a computer program that defines the (graphic) production.

purely visual, hybrid (text and visual) systems, and others (such as the form editor languages).

However, in all of the categories, the different semantic abstractions proposed by the languages (data, control, procedural) can have different representations, some being textual and some being purely graphical.

Also important are *programming by example* systems[65] (also called programming by concrete demonstration), which –regardless of the graphical representation chosen– allow programmers to exemplify the desired behavior, having the system automatically infer the general rules.

Even if the premises of visual programming are true –for human reasoning is visual– textual programming remains the undisputed king. This is partially due to the *scaling-up problem*: making visual programming languages suitable for solving large programming problems often requires to sort the same complexities (or even new ones) that visual programming tried to simplify [19].

What remains true is that visual programming often provides a gentler learning curve. Aiming to help artists (initially almost exclusively musicians) many visual languages were designed for artistic production and are very popular in artistic environments.

Patchers

Visual programming languages that explicitly represent the data flow through graphs (nodes plus arcs) are commonly referred to as *patchers.* Among those, we can find three languages designed by Miller Puckette[66]: *Max*, *JMax* and *Pure Data* (also known as *Pd*), all sharing *Max* as a common root.

These languages, especially *Max*[67] (named after Max Mathews[68]), with its modules *MSP* (named after Puckette) and *Jitter* for digital signal

[65] Some examples of programming languages of this kind can be found at http://web.media.mit.edu/~lieber/PBE/PBE-Examples.html.

[66] Miller Smith Puckette is a preeminent figure in the creation of interactive computer music systems. He holds a B.S. in Mathematics from MIT and a Ph. D. in Mathematics from Harvard. Dr. Puckette was a member of the MIT Media Lab from its inception until 1987, then a researcher at IRCAM, and now is the associate director of the Center for Research in Computing and the Arts as well as a faculty member at the University of California, San Diego, USA.

[67] What we now know as *Max* was originally called *The Patcher*, a window for the *MAX* system by Koechlin et al. [104].

[68] Max Mathews was a pioneer in computer music. He directed the Acoustical and Behavioral Research Center at Bell Laboratories from 1962 to 1985. He was Scientific Advisor to the Institut de Recherche et Coordination Acoustique/Musique (IRCAM), Paris, and is currently Professor of Music (Research) at Stanford University. He holds a

processing and video manipulation and generation, are the most used art-oriented programming languages[69].

Max was originally oriented to music production (today its many libraries make it apt for video, control and many other art-related tasks) with a bias to performative real-time use over compositional use, and was an attempt to recreate the modalities of a patchable analog synthesizer [105] within a programming language.

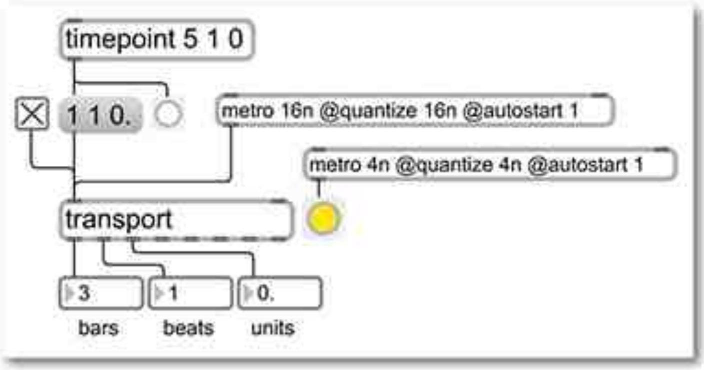

Figure 12 - A Max/MSP 5.0 patch (image courtesy of Cycling'74[70]).

> *The Max paradigm can be described as a way of combining pre-designed building blocks into configurations useful for real-time computer music performance. This includes a protocol for scheduling control and audio sample computations, an approach to modularization and component intercommunication, and a graphical representation and editor for patches. These components are realized differently in different implementations; and each implementation also offers a variety of extensions to the common paradigm. On the surface, Max appears to be mostly concerned with presenting a suitable "graphical user interface" for describing real-time MIDI and audio computations. However, the graphical look and editing*

Silver Medal in Musical Acoustics from the Acoustical Society of America, and the Chevalier dan l'ordre des Arts et Lettres, Republique Francaise. He wrote the computer program MUSIC I, which gave birth to the field of computer music performance.

[69] Although we were not able to find any study on the share of the different art-oriented programming languages, it is easy to see *Max* prevalence in the field. Puckette himself acknowledges its popularity by referring to the "wide use" of Max [106], or the community-written encyclopedia Wikipedia stating that "Max is widely regarded as the lingua franca for developing interactive music performance software" [137].

[70] David Zicarelli's company, producer of *Max/MSP* and *Jitter*, among other art-oriented software tools.

functions in Max aren't really original at all, and most of what is essentially Max lies beneath the surface.

Miller Puckette, Max at seventeen [106]

The *Max* family members (see Figure 12) are, then, two things: first – and more important, according to Puckette– real-time data-flow music and video oriented systems. Second, they include also a visual patching graphical interface, which is used to specify this dataflow (that is, to program it) and to visualize and debug it.

Although the first aspect is undoubtedly needed for the software to be able to perform according to what it is expected from a music or video performance-oriented software, it is the second one what has played the most fundamental role in patchers' extreme acceptance: "the question of making software systems which were really usable by non-computer scientists was addressed by the Max program" [105].

Together with this usability achievement (which evolved as an artifact of the original hardware devices, such as synthesizers, that they were based on), patchers usually provide their users with a rich set of objects that perform tasks in their area of work.

This is, as always, a double-edged sword as while they provide an abstraction to some of the difficulties, also generate some homogenization of the production.

A clear example is Netochka Nezvanova's *nato.0+55+3d*, a set of objects for *Max* that grew very popular and generated a great amount of very-similar, equally glitchy production from a lot of artists from all over the world.

To show *nato.0+55+3d*'s popularity we quote Cycling'74 (the company behind *Max/MSP* since 1998) in 2001: "The greatest significance of NATO.0+55 is that it has transformed Max from a piece of wonderful toolkit software, into a viable authoring environment, virtually overnight. Before NATO.0+55, Max was useful. After NATO.0+55, Max is indispensable." [91]

The popularity of *Max/MSP* and *Pure Data* introduced a culture of patching among artists, and several others languages have appeared, sometimes aiming to a specific platform or a specific artistic area.

Among the most popular are *Quartz Composer* and *VVVV*. *VVVV* is a Microsoft Windows-only toolkit for real-time video synthesis that allows its user to perform several video-related tasks, such as multi-projector setups, loading of 3D models, shader programming, physics simulation, video analysis, etc. [130]. Apple's *Quartz Composer*, on the other hand, is a Mac-only environment that gives access to the core audio and video technologies of Mac OS X [5].

Designing by numbers

> *Instead of the motor skills (of traditional art), today's digital designer must develop an awareness of the many capabilities and sequences of interactions in the continuously growing set of pre-packaged digital tools. In other words, skill in the digital sense is nothing more than* knowledge*, and the reality is that we implicitly glorify rote memorization as the basis of skill for a digital designer.*
>
> *The true skill of a digital designer is the practiced art of computer programming, or* computation.
>
> John Maeda – Design By Numbers *[75]*.

Although *Max/MSP* can still be considered the most successful of new media art's computer programming languages, not all the art-oriented programming languages are visual. As a matter of fact, with the popularization of languages like *Processing* and frameworks like *OpenFrameworks*, standard, textual programming has experienced a notorious growth in popularity among artists.

Besides its undoubted power, one of the main reasons for this growth is the work performed at the Aesthetic and Computation Group[71] of the Massachusetts Institute of Technology (MIT), directed by John Maeda.

Maeda's approach to design and computing, according to MoMA's Paola Antonelli, "paradoxically has a hands-on, almost Arts and Crafts feeling. His approach to computer graphic design is not different from an approach to wood carving" [75].

The conviction that, to broaden the frontiers, to escape from the constrictions imposed by computational tools, and to create new expressive languages, the appropriation of technology behind such tools is needed, have led Maeda's team to create a simple programming language, *DBN*[72] (acronym for *Design By Numbers*).

This language, which actually is both a language and a programming environment, was explicitly designed to "teach the idea of computation to designers and artists" [76], although it did not reach the popularity its author expected.

Instead, perhaps *DBN*'s greatest accomplishment is that two of Maeda's students, Ben Fry and Casey Reas, who were the "voluntary

---

[71] http://acg.media.mit.edu

[72] http://dbn.media.mit.edu/

caretakers of the DBN system, felt that they had the right design for a new solution" and designed another language oriented to designers and artists, *Processing*[73].

Processing, in the vein of *DBN*, was originally designed to promote software literacy among designers and artists. However, it evolved into a professional –but still accessible, open source and free– extremely popular tool.

In effect, *Processing* is used at hundreds of schools around the world, at universities, and graduate programs. Also, "tens of thousands of companies, artists, designers, architects, and researchers use Processing to create an incredibly diverse range of projects" [103].

It has also been awarded multiple times, having received the Golden Nica award from Ars Electronica and the Interactive Design Prize from the Tokyo Type Director's Club in 2005, The Cooper-Hewitt National Design Museum included Processing in its National Design Triennial, and Fry and Reas received the 2008 Muriel Cooper Prize from the Design Management Institute.

Processing's popularity can be explained by a multiplicity of factors, many of them are a direct consequence of Processing programs ("sketches" in Processing lingo) being compiled onto Java source code.

The main reasons for *Processing* extreme acceptance are:

A simplified syntax that does not hinder power
Processing is relatively easy to learn and use, while offering –unlike *DBN* and other dedicated environments– unrestricted programming power.

The gentle learning curve of *Processing* is obtained in part thanks to its simplified but scalable syntax: while inheriting all the expressive and modeling power of object-oriented languages, *Processing* also allows for procedural programming.

Novice users can program in an entirely procedural manner (while still being able to access all of the native and libraries' objects), and more experienced users, on the other hand, can adopt the object-oriented programming paradigm, while still programming in *Processing*.

Likewise, the language and environment (see Figure 13) isolate the programmer from many of the complexities of programming: not only the syntax is simplified, but it also offers a very rich set of libraries that handle many of the problems that commonly surface in art-oriented coding. There exist official libraries for video, networking, serial communication, exporting to several formats (PDF, DXF, SVG, XML),

---

[73] http://processing.org

sound synthesis, MIDI, controlling *Arduinos*, 3D (with an OpenGL wrapper via JOGL) and 2D animation, computer vision, math and physics simulation, and many more[74].

A consistent, step-by-step, online documentation

Processing.org, the language's web site and virtual community center, offers[75] several tutorials and examples that introduce the primary elements of computer programming, the different topics and the more important libraries.

The basic examples cover the following topics: structure, image, input, form, shape, transform, data typography, color, arrays, control, objects, math and web. More advanced tutorials and examples are found in the section "topics".

There are also several official books that help with Processing learning: besides Reas and Fry's *Processing: A Programming Handbook for Visual Designers and Artists*, Ira Greenberg and Daniel Shiffman have written their own books, with the latter specifically aiming to the programming illiterate.

A custom, simple, programming environment

As we have mentioned, *Processing* is more than a language as it also includes an Integrated Development Environment (IDE). Even if every major language offers IDEs (many of them are free, like code::blocks[76] for C++ and Eclipse[77] for Java), *Processing*'s own IDE is designed for maximizing its easiness of use (see Figure 13).

The IDE also includes all of the examples that we have previously mentioned, encouraging a hands-on approach to novice users and offering many pre-created solutions helpful for both novice and experienced programmers.

Multiple platforms, including web

Inherited from Java, *Processing* programs can be natively run from within a web browser. This permits to display on the web a source code together with its output, and therefore there are thousands[78] of web pages showing thousands of programs.

---

[74] For a comprehensive list, please see: http://www.processing.org/reference/libraries/ Also to be seen are the spin.off projects mobile (http://mobile.processing.org/) and hardware (http://hardware.processing.org/)

[75] See http://processing.org/learning/index.html

[76] See http://www.codeblocks.org/

[77] See http://www.eclipse.org/

[78] As of Feb 12, 2009, the standard signature "built with processing" shows "about 80.000" results in Google's search engine.

In the same vein, *Processing*'s web site examples do show they result together with the source code.

Processing's IDE runs natively in Apple Mac OS X, Microsoft Windows and Linux, and, also inherited from Java, creating applications for those targets is immediate, as they are Java Virtual Machines for them.

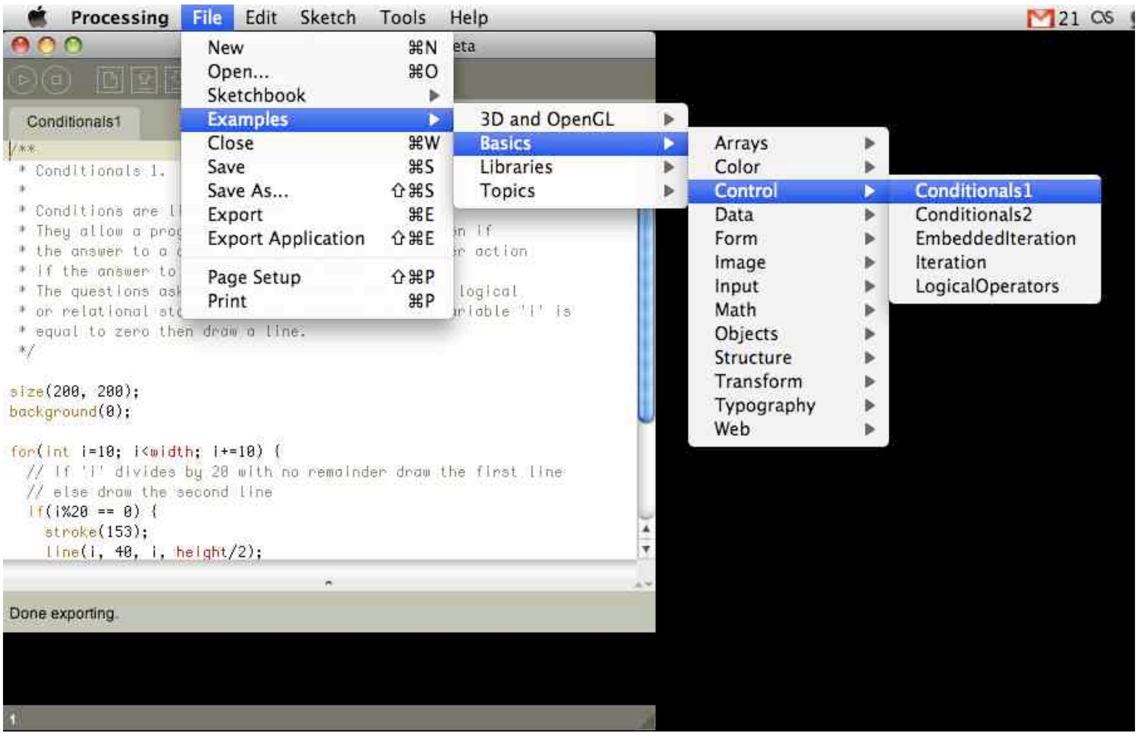

Figure 13 - A screenshot of the Processing IDE, showing menu for loading the included examples.

Easiness to migrate to other (art-oriented or not) programming languages

Processing can be used as a gentle entry door to computer programming: thanks to the strong resemblance of its syntax to other languages such as C, or, especially, Java, users feel not only that learning *Processing* constitute an end on itself (for it is a tool used in world-class computer art), but also that they are not trapped in it. It explicitly leverages software literacy, or, in other words, explicitly enables new media appropriation.

Also, migrating from *Processing* to direct Java programming is an often-transited and well-documented road. Java users can still use all the *Processing* libraries and it becomes a programming framework (very similar to *openFrameworks*).

An active community and an open-source model

Finally, both an effect and a cause is the extremely active and numerous users' community that not only discusses and helps in Processing.org's forums but also participates in the creation of the language itself.

In addition, *Processing*'s compiler and environment are open source projects and have already benefitted from the collaboration of many individuals.

> *I also think it* [Processing] *encouraged a slightly superficial view of computational design by quickly gaining cult status amongst people never been exposed to programming before. I think it's dangerous and a sign of crisis if every recycled L-System, Neural Network, Wolfram automata or webcam tracking experiment automatically is considered art (by their authors), simply because it's been "(Re)Built with Processing.*
>
> Karsten Schmidt (a.k.a. toxi) [116]

Processing's popularity has had noticeable effects. Schmidt addresses, in the above quote, the effluence of projects built with *Processing* that create and re-create new media artworks, yet offering nothing but the (valuable) finding of their author's own abilities.

Among the consequences of *Processing*'s popularity, we can mention *openFrameworks*[79].

OpenFrameworks, created by Zachary Lieberman and Theodore Watson, is a very popular C++ software framework[80]. It was strongly based on *Processing* from its inception, and offers a similar set of features while profiting from C++'s performance (which is often perceived as better) and from the enormous number of C++ libraries.

It is perhaps also worth to mention a third popular programming language among artists: Adobe Flash's *ActionScript*. *ActionScript* (which, like JavaScript is based on ECMAScript) allows Flash designers to program within Flash and has been used in multiple online art pieces.

But popularity is not the only variable to take into account when discussing the relationship between computer programming languages and art. We will now discuss two less-popular languages that show a different approach, which questions this relationship.

Designing with less numbers

Almost every traditional programming language that is used to produce an artistic output requires a behavioral deconstruction of the intended oeuvre.

---

[79] See http://openframeworks.cc

[80] It should be noted that while Processing is both a programming language and an application framework (if instantiated from pure Java), OpenFrameworks is a pure C++ application framework.

This is the traditional relationship between programming and art; the artist specifies a sequence of actions to be carried on by the computer by means of an imperative[81] or a visual programming language.

There are, however, some examples that show other possibilities by re-signifying the programming gesture. We will see two of those: *C5* and *Piet*. The former, adds expressivity to type declarations allowing for declarative programming within its graphical library and the latter somewhat shifts the aesthetic relationship from the programmer to the language designer.

C5

C5, by Juan Cabezas[82], is a superset of the C programming language that extends its type system, allowing for the definition of types in the form of dependent pairs: a couple where the type of the second member of the pair depends on the value of the first ones [20].

With this extension *C5* becomes a generic programming framework, where generic functions (functions that work regardless their parameters' types) can be defined. Furthermore, it allows type declaration to become a very expressive member of the program, even to the point that "type declaration can be the main code of a program" [20].

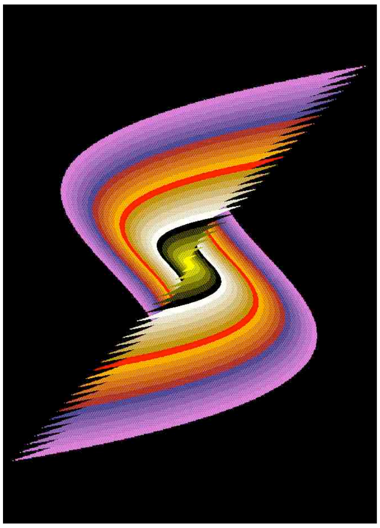

Figure 14 - *The flying S*. An image generated in *C5* by Cabezas. Image courtesy of J. Cabezas.

[81] Other programming languages paradigms have been used in art-oriented programming, most notably agent-oriented and functional programming, although to a lesser extent. An interesting and recent effort of extending Processing with functional programming in Scala (an object-oriented functional language), which (as of February 2009), can be seen at http://hipstersinc.com/blog/2008/1/23/scala_and_processing/.

[82] See http://www.fing.edu.uy/~jcabezas

Although these extensions are not directly related with any artistic interest, the included graphics library does relate *C5* with the visual arts in general, and *Universalismo Constructivo* (Constructive Universalism) – Joaquín Torres García[83]'s constructivism– in particular.

Constructive Universalism was Torres' aesthetic movement, where he conceptualized artworks as constructions composed of drawings that represent the idea of something (rather than the thing itself), and an ordering of such images to establish a rhythm in the work as a whole [63].

Torres, then, defined his art movement based on two concepts, structure (in order to give a unity to the construction) and abstraction (ideograms representing thing and ideas in order to use universal representations) [108].

C5's graphic library, called the Oriented Port Machine (OPM), proposes a deconstruction and a formalization of Torres' pictorial methods, as follows:

- construct the color planes of the page.
- construct a rectangle structure representing the image structure.
- for every rectangle of the structure, stamp an ideogram or construct a structure representing the rectangle image.
- continue this structuring process until the desired image is obtained. [20]

What makes *C5* of particular interest is not the appropriation of an aesthetical framework, or the constructive approach[84], but the rather radical shift expressive means of modern languages, from functional or object construction based to one based on types.

In effect, the principal function of *C5*'s graphic library, opm_image_cons(), receives a dependant pair as argument, and its behavior –its graphical output– is determined by the *type* of the parameters.

This allows *C5* users to actually program by defining types instead of the traditional specification of operations to be carried on variables of known types.

[83] Joaquín Torres García (1874 – 1949) was a Uruguayan painter and sculptor (probably being now Uruguay's most celebrated artist).

[84] Numerous examples of art-oriented developments can be found, from "Jackson Pollock generators" to Photoshop plug-ins that modify photographs to resemble oil paintings. Also, constructive approaches are popular, with constructive solid geometry being a well-known example.

Piet

Piet is a programming language by David Morgan-Mar, designed so that its source code looks like abstract paintings [87] (see Figure 15).

A program source code in *Piet* is a graphic made up of the recognized colors. The *Piet* interpreter reads the color blocks starting by the upper-left corner, with the sequence of reading determined by the source code itself.

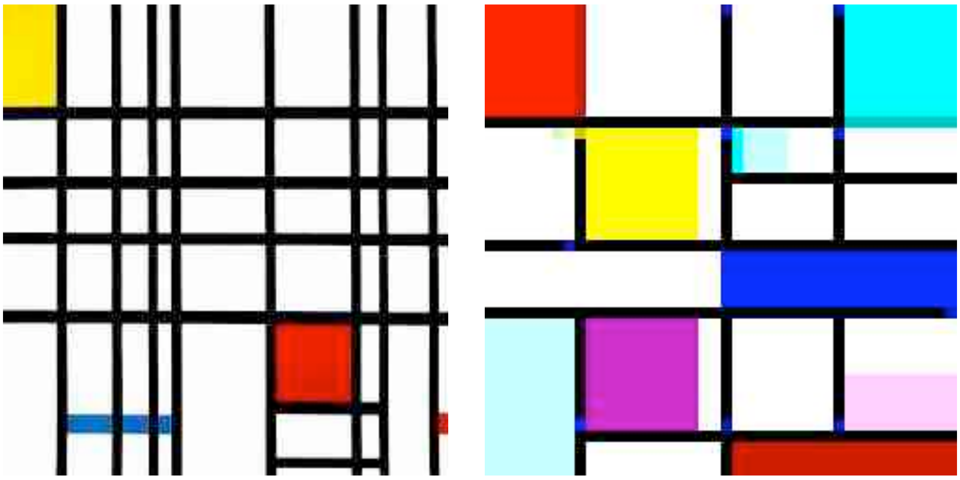

Figure 15 - Left: *Composition with Red, Yellow and Blue*, Piet Mondrian, 1921. Right: A program in Piet that prints the word "Piet", Thomas Schoch, 2006 [118].

Piet source code uses twenty colors, of which eighteen are related cyclically through a lightness cycle and a hue cycle (the other two being black and white). A single stack is used for data storage, and the operations are defined by the transition of color from one color block to the next as the interpreter travels through the program.

What interest us are not the details of this esoteric programming language, but that by creating a language whose source code does look like an abstract painting, the artistic fact shifts from the user of the programming language to its creator thanks of the explicit aesthetic dimension added to the source code.

Piet programmers are *forced* to create abstract paintings without any, a priori, artistic intention, and *Piet* programs could be used to perform any[85] computation.

It is clear, though, that nobody would choose *Piet* as a programming language if not *because* of the aesthetic appeal of its source code;

[85] Although it is suspected, it has not been formally proved that *Piet* is a Turing-complete language; therefore, it is not really possible to claim that any computation can be performed. [36]

therefore, *Piet* subverts the very reason of existence of programming languages.

Programming art

Creating an aesthetic design of a programming language is consonant with the already stated idea of *composing technology*: technological production is not only aesthetical relevant but constitutes a raw medium for art creation.

In the examples we have seen, the relationship between art and computer programming is faced in different ways.

Traditional –textual or graphical– programming languages attempt to provide artist with a refined, complete and accessible tool to create their artworks.

All these languages share a common conception on *how* the programmer codes the desired behavior.

C5, on the other hand proposes a different approach to aesthetical computing: using an artistic movement to shape the methodology of coding.

The structural pairing between code and design acquires here a new dimension, as the proposal is centered on the creative process rather than in the designed object.

Piet, finally, proposes an aesthetic dimension in both the design of the programming language and its source code.

Even if it is true that computer programmers take decisions in terms of the code's aesthetic, this aesthetic rarely is visual, and tends to respond to the behavioral and modeling design of the program.

4 EXAMPLES

In this chapter we show some artworks created within the framework of this thesis that exemplify the discussion of the first chapters: our conceptual framework of new media art, media appropriation and the relationship between new media art, computer programming and human-computer interaction.

As the first two artworks, Puzzling and YARMI, operate in augmented-reality space, we will start with a short discussion of augmented reality itself.

Augmented reality

> *Virtual reality is like, practically, totally real, but not.*
>
> Lois Kaiser, character in Robert Altman's film *Short Cuts*, 1993.

Augmented reality (AR) –also sometimes called mixed reality– is the juxtaposition of virtual (computational) images and *real* ones –where real images are those acquired by some mechanism whose transduction we have learned to ignore– together with the provision of some interaction scheme for its users.

AR techniques rely strongly on the assumption of reality –what Vilém Flusser called the *technical image* [43]– and also on the self-describing computational interfaces and interaction styles that the system proposes with the addition of virtual images.

Flusser catalogs the technical image as *applied scientific text,* that is, the concretization of a model of reality by the visual apparatus. The technical image is then an indirect product of text that, due to its character apparently non-symbolic, objective, becomes dangerous, as some tend to forget that they are, always, extremely abstract.

Perhaps it is worth to remember that, as W.J.T. Mitchell argues, all media is mixed media [85].

Rendering synthetic images over real-time acquired images can be also seen as a type of multichannel cinema (picture-in-picture), which adds the notion of spatial montage to the traditional temporal one, bringing to cinema the aesthetics of other realms such as video games or military graphical interfaces [110].

Videogames, for instance, tend[86] to not use film-like cuts as a narrative resource (except for the non-playable cinematic sequences that

[86] There have been numerous attempts at interactive storytelling that emphasize the narrative component of videogames. These "narrative videogames" (sometimes

sometimes are included for storytelling, and use the traditional language of film [12]), but, instead, use multiple simultaneous channels.

These parallel channels, like the borrowed-from-the-military first person shooters' head-up displays (see Figure 16), superimpose game information (for example, the character's energy left) improving communication bandwidth, playability, and providing new ways of interactivity; although sometimes also hurting the players' suspension of disbelief and their immersion into game play [12; 138].

It is worth noting that similar resources have been used –very succinctly– in cinema, being an easy example the common split-screen telephone conversations [110].

Figure 16 - Left: A FA-18 Hornet aircraft's HUD (image on the public domain). Right: *Tom Clancy's Ghost Recon Advanced Warfighter 2*'s HUD (image courtesy of Ubisoft)

There are also examples of AR projects that focus on creating or re-creating traditional film narrative, with a twist. Examples of this are MacIntyre, Moreno et al.'s AR setups aiming to immerse spectators *into* the narrative by changing their point of view and providing an interactive approach to the film [73][86].

Art pieces that use augmented reality suppress some stages traditionally present in art creation. More exactly, AR renders (more) implicit some aspects that in traditional (cinematographic) setups are explicitly presented, such as the design of the scenario, the montage or the sound recording [109].

The most common AR-based art pieces are the often called "magic mirrors": camera-and-projector setups where the camera points to the spectator and the projection shows, in real-time, an affected image. Many examples of this can be found –mirrors were the first interactive

referred to as interactive movies, with Dragon's Lair probably the most famous example) are based on a structure where users must discover the winning patterns of choice. The popularity of this type of videogames has declined notoriously, probably because the feeling of interaction has responsiveness as its main factor of success [101]. Art-oriented interactive movies have also appeared since the beginning of new media art, and still conform an active area [100].

art pieces[87]–, and appear so frequently and systematically that they have even been categorized as a design pattern [35].

Figure 17 - A screen capture from the augmented space of Julian Oliver's art installation *levelhead*.

A particularly interesting installation, which was of great inspiration to us and also has attracted much attention and acclaim, winning an honorary mention at Prix Ars Electronica '08, is *levelhead*, by Julian Oliver (see Figure 17).

According to its author, *levelhead* is a spatial memory game that uses a hand-held solid-plastic cube as its only interface. The cube (together with the user) is shown on-screen, augmented by rendering its (virtual) interior on its faces, as if they were windows to cubic rooms.

In one of these rooms is a character. By tilting the cube the player directs this character from room to room in an effort to find the exit.

There are three cubes (levels) in total, each of which are connected by a single door. Players have the goal of moving the character from room to room, cube to cube in an attempt to find the final exit door of all three cubes. If this door is found the character will appear to leave the cube, walk across the table surface and vanish. The game then begins again. [94]

As Rhodes notes, it is worth pointing out that Oliver's states that the only interface of *levelhead* is the cube, because the projection as mediated reality is so common that is not considered part of the interface [110]. Again, the illusion imposed by the technical image shows its strength in shaping what is naturally considered as real.

[87] I heard this (here paraphrased) sentence on a talk by Chinese artist Jay Yan in the festival FILE007, Sao Paulo, referring to standard, traditional mirrors.

Fiducial-based augmented reality

Augmented reality always involves computer vision, although, unlike it, does not aim to solve the AI-complete problem of vision, but attempts to extract the relevant features that allow the real-time augmentation of the images. [8]

For some AR applications, it is then acceptable to place fiducials –easily recognizable markers– into the scene to ease the vision problem. These fiducials may be LEDs, colored dots, or other type of special markers, for which the location or pattern is assumed known [8].

Among the fiducial-based augmented reality solutions, some libraries have been often used in art installations, such as the C++ libraries ARToolkit[88], ARToolkitPlus[89] –which we used in our examples– and ARTag[90], all based on fiducials with recognizable patterns (see Figure 18).

Puzzling[91]

Puzzling is an interactive art piece presented as an art installation, a cinematic manipulation tool, a game, and a toy.

It proposes the exact opposite of MacIntyre or Moreno experiments: in it, the traditional cinematic material provides the computational images that are rendered onto the "reality".

There exists at least one previous experiment on the same vein: G.A. Rhodes' *52 Card Psycho*[92] (see Figure 19)[111]. In this installation, the fifty-two individual shots of the famous shower scene of Hitchcock's *Psycho* (1960) are superimposed onto a deck of fifty-two cards imprinted with a unique fiducial identifier.

Rhodes' setup, while inspiring, does not question the basic cinematic unity –the frame– and provides only one interaction scheme with the art piece, relying strongly in the incongruence provided by the mundane gestures of card playing. This association (between cards and shots) does allow "a material interaction with the 'cinema screen'", where users can create "spreads of time", recomposing the cinematic flow.

---

[88] http://www.hitl.washington.edu/artoolkit/

[89] http://studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus.php

[90] http://www.artag.net/

[91] In this work participated undergrad student Ernesto Rodríguez.

[92] http://52cardpsycho.com/

We find particularly interesting that *52 Chards Psycho* permits laying the fifty-two scenes in unison, allowing users to create spatial compositions that show the chosen scene simultaneously. Yet, this maintains the frame as basic unit, while losing the narrative's intrinsic dimension of time.

Figure 18 - An ARToolkitPlus fiducial (image from ARToolkitPlus' web site).

Our work, on the other hand, attempts for a deconstruction of the film's frame, allowing for its deconstruction and montage (both temporal and spatial), also positioning itself as a direct manipulation tool of filmic material *and* a video game of sorts.

This questions the spectators' role, offering them the possibility to choose *how* to relate with the installation and the filmic material.

The installation

Puzzling's base idea is to deconstruct the frame –in augmented reality space– breaking it down into equally sized portions, with each portion superimposed onto tangible pieces.

Each piece is tagged with a unique identifier (see Figure 20), creating a jigsaw puzzle of sorts.

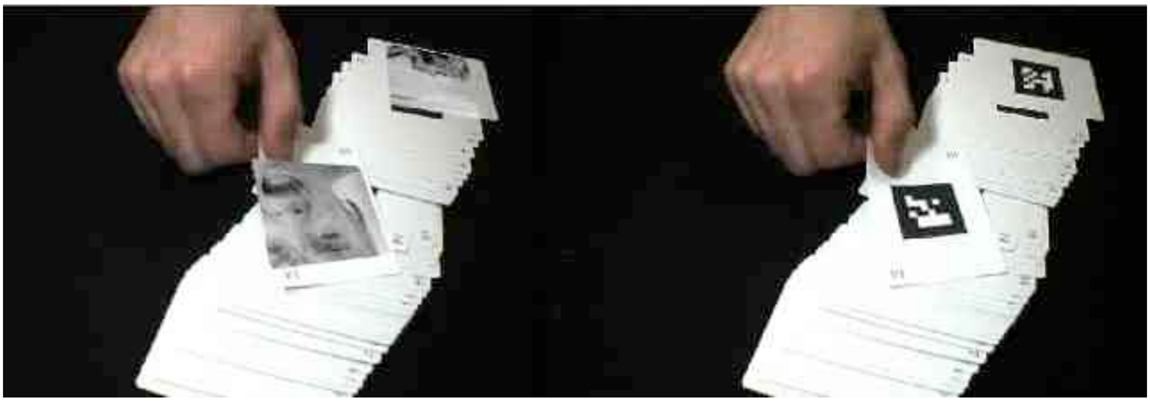

Figure 19 - G. A. Rhodes' *52 cards Psycho*, an Augmented Reality-based installation. Left: the augmented image with the movie's scene superimposed on the card. Right: the same image without augmentation, showing two fiducial markers (image courtesy of G. A. Rhodes).

Users are presented with the tagged pieces on a table (its number can vary, but we usually use nine, as we found it provides the best results), which is viewed by a mounted zenithal camera.

Accompanying the table is a real-time video projection[93], in front of the users, showing the *screen space*, that is, the images acquired by the camera with the synthetic images superimposed (see Figure 21).

In the screen space the augmented reality is shown: there coexist the table, the portions of the frame (of course, still playing the film), and the hands of the users. A hybrid space appears where the direct manipulation of the deconstructed cinematic material is possible.

Interaction

Users can interact with the cinematic material in different ways. First of all, and probably most important, is the spatial reconfiguration of the frame: by just moving the wooden pieces the frame disposition changes, allowing for new configurations. It is possible, also, to map more than one frame (or portions of different frames) into the pieces enabling users to create new compositions, distorting the existing or creating new narratives, potentially very different from the film's original story.

[93] We have chosen a projector-based display only because of its size, but any kind of modern display would suffice.

Figure 20 - The camera acquires the pieces' positions and the system then renders the augmented image with the decomposed frame.

But *Puzzling* also allows for the control of different parameters of the reproduction. For example, it is possible to alter the speed of the video reproduction rotating the pieces. The frame then becomes unsynchronized, with portions speeding –and soon showing images of the future– or relenting, showing images of the past of the film[94].

We found interesting one of the setups, where the image cannot be spatially coherent (in the sense of recreating the original disposition) and synchronized at the same time. If the geometry is recreated, then the frame is temporally dissonant and vice versa. This creates a tension for the users to resolve: what is to be favored, temporal or spatial coherency?

Interaction modes

Puzzling, with its very simple underlying idea, offers a very rich interaction scheme, with many possible gestures and effectively creates the idea of direct manipulation of the filmic material.

We have found that depending on the users' intention when using the installation, it becomes a game *or* a performance tool (and, stretching the definition it can be seen as a film analysis or edition tool).

[94] Other variables, aside from the temporal, can be easily mapped to spatial variables. For example, a color component to the rotation or the position on a given axis.

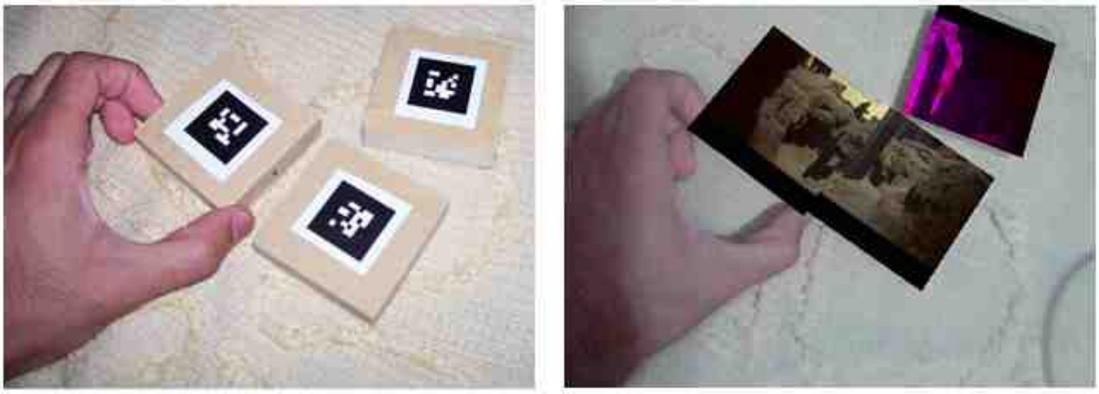

Figure 21 - Left: three tangibles marked with fiducials. Right: the actual augmented reality space (screen space), with the moving images superimposed and two pieces consistently placed.

The setup can suggest the mode of interaction. For example, if *Puzzling* is to be used as a game where the users' goal is to recompose the frame (again, like an animated jigsaw puzzle), *Puzzling* can add a visual clue when two or more pieces are together in their correct position (see Figure 21. In the right, one of the three fragments is tinted showing that its position is not consistent, the other two are rendered with their original colors).

When *Puzzling* is used as a game, users focus almost exclusively on the detection of the internal coherence of the frame, easily forgetting its narrative possibilities as the actual content only matters to them as information to solve the task.

It is worth noting that even when the frame is decomposed in a few pieces (our setups usually use nine), the task of rearranging the frame can be very difficult.

Other similar games can be easily imagined and added to *Puzzling*; we have experimented with tasks that comprise using several frames, arranging them spatially and chronologically, or tasks where users need to find the similarities between different scenes.

In the first case, we used footage from *2001, A space odyssey*, and on the second one we used material from *Short Cuts* (see next section for a discussion on the filmic material).

But the installation could be also an expression tool (or a toy, i.e. a game with no intrinsic goal, apart from enjoying using it), where the spatial and temporal arrangements of the cinematic material become a new way of expression, a new narrative orthogonal to the film's intrinsic one.

This approach to using *Puzzling* is very similar to live cinema's one: the real-time editing and composing of pre-recorded footage becomes a visual art performance.

Cinematic material

While, as it is inferred from the previous section, *Puzzling* can be used as a tool to play with any cinematic material, our installation uses two referential movies: Stanley Kubrick's *2001: A Space Odyssey* and Robert Altman's *Short Cuts*.

2001: A Space Odyssey

Stanley Kubrick's 1968 iconic film, considered by many as one of the best films ever made [17], contains several scenes instantly recognizable by a great amount of people.

Partially responsible for this recognizability are the film's extremely rich visual language, and distinctive pace and music (although *Puzzling,* is a silent work). In effect, its visual richness is so compelling that only few scenes are actually driven by dialogues.

This distinct visual density, when deconstructed in *Puzzling*, give its users strong cues on how to reconstruct the original frame, while at the same time allowing for new configurations (both spatial and temporal) that effectively turn *Puzzling* not only into a fun game but also into the aforementioned expression instrument.

While some of the scenes are extremely easy to reconstruct, some others –like those depicting weightless sequences– lack, for example, directional cues (there is no up or down), allowing for different semantically sensible reconstructions.

Our main criterion for choosing scenes was to focus on those displaying geometrical patterns (e.g. the early men's scene on the first act), specifically those that allow for multiple different, yet sensible, visual configurations.

Short Cuts

Where *2001...* provided *Puzzling* with slow paced narrative that somewhat encouraged its users to search for new in-frame configurations, Robert Altman's *Short Cuts* (1993) narrative speed and parallel storytelling, when deconstructed with *Puzzling*, urge its users to re-combine and create new lectures of the stories or even new stories altogether.

*S*hort Cuts "consciously refuses to single out any particular character" [7], showing them all –twenty-two arranged in nine groups– and their (independent but sometimes intertwined) stories with equal importance.

These stories are presented in an already deconstructed style: hundreds of segments ending with a cut ("never a dissolve, never a fade to black, never a wipe. Cuts only" [10]), linking the themes from the stories, which –most of the time– are concurrent.

But what makes *Short Cuts* extremely apt for *Puzzling*, are the techniques used by Altman to interlink the stories. In the words of David Balcom

[10] "The film offers associations between characters, and between mini-sequences, that an active viewer can construct along with the stories they see on the screen. There are subcurrents flowing in the text that link thematic material throughout the film. In fact, most themes are linked in more than one direction (from material that preceded it, and toward material that later recalls it), and to more than one theme." Several scenes reappear in different forms providing formal links between the stories, with different characters in similar positions or situations, providing "a tremendously rich body of associations at work in the film that simple narrative analysis cannot account for".

But perhaps *Puzzling* users can.

Implementation

The installation consists of the following elements: a table where users can manipulate the pieces while looking at the output video, a number of tagged wooden pieces, a camera that views the table and a projector or screen that shows the augmented video to the user and the audience.

Puzzling is, basically, a work of software. All the hardware used (apart from the pieces of wood), is standard, off-the-shelf, only requiring a webcam and a reasonably powerful computer.

The installation's software is platform independent, being able to run in any modern operating system (it has been deployed on both Microsoft Windows and Mac OSX).

We coded it using the already mentioned software framework OpenFrameworks (a collection of libraries that help enormously with many common tasks of multimedia applications) and the ARToolkitPlus library, "a software library that can be used to calculate camera position and orientation relative to physical markers in real time".

It is important to note that both the camera and the light sources need to be put in adequate positions and correctly calibrated or the Augmented Reality illusion will (mostly) go wrong.

Puzzling's software cuts the footage into a given number of equally sized pieces, and automatically maps each piece to a marker that appears on screen.

When a marker is identified, the system overlays the corresponding video, applying the same geometrical transformations recognized in the marker, completely substituting it in *screen space*.

Conclusion

Using *Puzzling* to manipulate the cinematic space shows two possible levels of representation of information, that is, two different and disjoint modes of user engagement; but in both cases, the direct manipulation of the footage questions the relationship between the viewer and the cinematic material.

In effect, the relationship between users and the narrative changes drastically, in one extreme, with users taking a playful role, the cinematic narrative is an accessory that provides coherence to the deconstructed bits of the film. The narrative itself is not important except for its cognitive density: it hints on the relationships between the different blocks.

Equally important is the approach of considering our installation as an ephemeral non-linear, spatially distributed tool.

Users' goals change. Not being playful, they can relate more to the film's content and interact with it. They are no longer spectators but became part of the film: their role and the film's role change.

It is worth noting that in all the examples in this sections presented the theoretical framework and media appropriation is very present. Besides the evident relevance of computer programming in *Puzzling* creation, some of the mentioned HCI techniques were used, namely iterative design, and user-centered design.

Puzzling's interaction was evaluated (albeit rather informally) and the it's responsiveness was identified as the most important factor affecting user's engagement.

And the product of this interaction is not more important that the interaction itself, its aesthetic: *using* the installation becomes an artistic performance, editing –in a way that we believe would satisfy Kuleshov's idea of edition's role– *becomes* the art fact (which is also consonant with Live Cinema's proposal's proposal).

But this art fact is an ephemeral one: each performance, each playing with our installation, our tool, could be unrepeatable and last only in the memory of the performers or their audience.

YARMI: an Augmented Reality Musical Instrument

The field of computer based musical instruments, while very active and vital, has produced several instruments that somehow lack playability or expressiveness in traditional musical terms.

While the composition[95] of new musical instruments can provide composers and performers with tools for new music and musical languages, the instruments produced are often too simplistic (like the rather naïve new incarnations of traditional step sequencers and drum machines), lack of playability (often due to the delegation of too many performative decisions to the instrument, not providing an effective fly-by-wire alternative), or are too different from traditional instruments

[95] See *Digital Lutherie* in chapter 2, *Human-computer interaction*.

(those with a social agreement on how they should sound, how they should be played or how they relate to others instruments output).

While the eclecticism of new musical instrument production is easy to note, also "the technological convergence is diminishing the identity of the music technology" [120]: in an ever-more technologically imbued world, the mere fact of using new technologies does not turn an instrument into something interesting.

Our design goals

YARMI's idea (and its name) emerged from a discussion in our lab: which are the aspects that make an instrument truly playable and not *Yet Another Ridiculous Musical Interface*? What design patterns can be applied to achieve playability, engagement, and musical sense? And also, if there is no social knowledge on the instruments' use, how can the public relate to the performance? And can the public decode the performers' gestures and relate them to the sonic output of the instruments?

In order to build an instrument that meets those expectations we decided to use two design patterns: (directly) mapping sound to physical objects, and traditional music sequencers.

A rough division of computer music control, based on the immediacy of the sonic response to the interaction, can divide controllers in sequencers and continuous, also called gestural, controllers.

Music sequencers provide a means to describe a sequence of sounds that are to be produced by the system, and play a central role in computer music creation [34].

Continuous controllers, on the other hand, provide a direct control of the sound being produced, allowing the performer to trigger sounds or modify some of its qualities, more in the vein of a traditional musical instrument.

Both design patterns are extremely important. While sequencers are the traditional tool construct digital music, direct manipulation approaches enhance users' engagement and (real-time) expressiveness.

In addition, both sequencers and direct manipulation gestures can offer a very gentle learning curve to the performers-to-be while being able to be easily interpreted (that is, to establish a correspondence between the gestures and the produced music) by the audience during a performance.

In effect, musical sequencers are standards that do not pose a metaphor but constitute a recognizable (and comprehensible) interface themselves.

In the same vein, physically based, direct manipulation interaction, constitute an established paradigm in tangible user interfaces, with the successful examples of the *Reactable* [60] (together with Oliver's *levelhead*, a very strong source of inspiration for us), *ISS Cube* [107], *Audiopad* [95], or *Block Jam* [92].

This interaction style allows users to *feel* that they are operating directly with the objects presented to them [93], also allowing for an easy deconstruction of the performance from the audience.

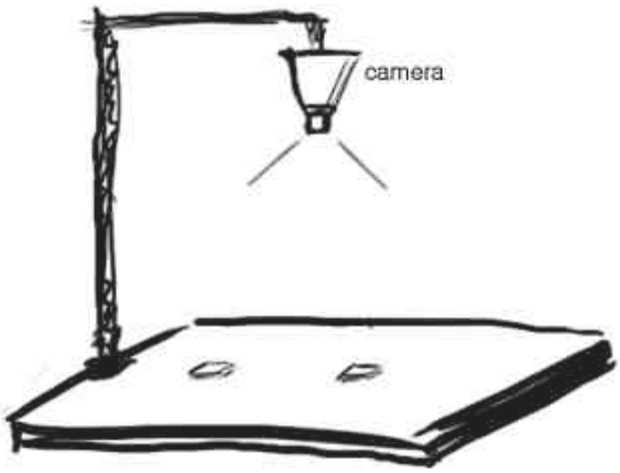

Figure 22 - A station's table with the zenithal camera mounted.

The instrument

YARMI is a collaborative musical and –to a lesser extent– visual instrument.

It was designed to offer tangible, direct, multi-modal, and multi-user interaction, with a shared (between the performers and the public) performance space with explicit visual representation.

Stations

YARMI is a multi-user, distributed instrument; or rather, an ensemble of synchronized instruments, operating under client-server architecture.

Each performer operates a *station* and *YARMI* is comprised of an arbitrary number of *stations* and one server.

A *station* consists of a table (or any flat surface) with a zenithal camera mounted[96] on (see Figure 22) and a visual projection showing the *screen-space*, an augmented version of the station's table.

96 We are designing a camera mount, so that any standard table can be used for a station. We hope this will come handy when performing at public spaces.

On each *station*'s table, users[97] can put tangibles –analogue to *Puzzling*'s wooden tokens with fiducial markers– that are recognized by the *station* and provide the only mean of interaction.

Setup

Each station has its own visual representation showing the augmented surface, which we call *screen space*, but the table remains with no augmentation at all: for both the performer and the audience, it is just a flat surface with some wood pieces on it.

The locus of attention of both the performers and the public is the *screen space*. This real-time projection of the augmented surface shall be set up so that the audience stands between the performers and the images, visible by everyone and providing an explicit representation of the performers gesture and the different *stations'* visual feedback.

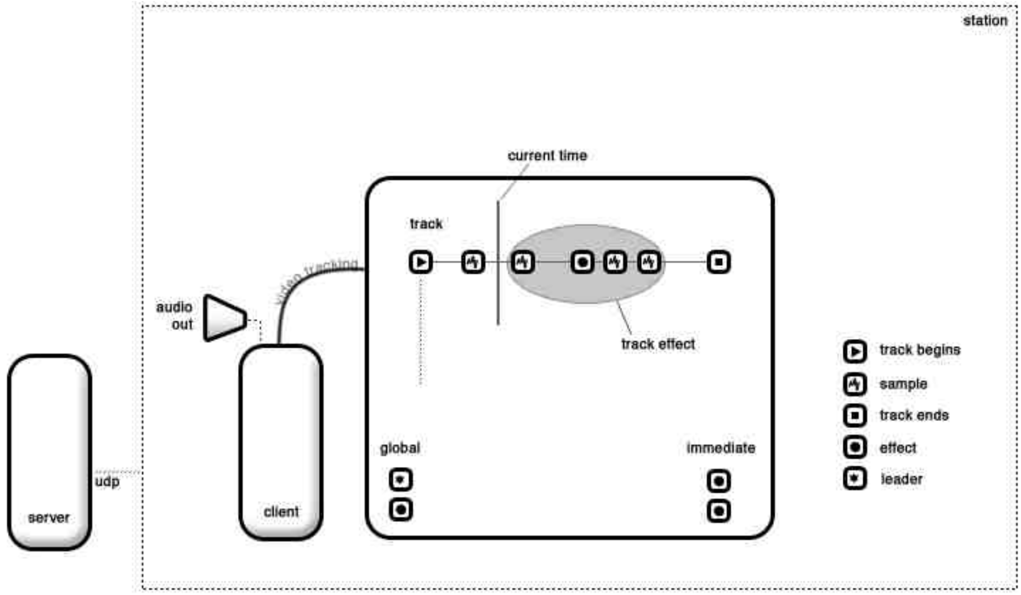

Figure 23 - Schematics of one *station*, with some aspects of the screen space superimposed, and *YARMI*'s server

Interaction

Each station is divided into three different zones named, *track zone*, *effects zone* and *global zone*, which we will now describe.

The *track zone* is an implementation of a multi-track sequencer, where users can create tracks and add samples and local effects to them.

To create a new track, the performer must add two tokens, one marking its beginning and one marking its end (see Figure 23).

97 Depending on the local setup it could be chosen to have more than one performer per station. Our design, however, is focused in the one-performer-per-station setup.

Once a track is created, users can add new tokens to it indicating samples[98] to be played, or local effects to be applied.

In every case, the rotation of the token controls the main parameter of the effect or the sample's pitch.

The *effects zone* (which we also refer to as the *immediate* zone) presents the likes of a sound effects machine. Each token put on it triggers an immediate response.

If the performer adds a token representing a sample, the station starts to play it, looping it as long as the token is present (token rotation always controls the sample's pitch).

If a token representing an effect is added, the effect is applied immediately to the *station*'s output, i.e. the mix of all its tracks and local effects.

If many effects are added, they are applied respecting the order in which they were added to the zone.

Finally, the *global zone* is the *settings* zone, where users can add tokens that modify the *station* or the ensemble behavior.

In-station synchronization

Being a multi-track and multi-user instrument, synchronization between tracks and between stations sometimes is fundamental to produce coherent music.

Each track is automatically synchronized so they all start playing at the same time, but, as they can have different lengths, the first track that is created in the leader station (see next subsection) –called the main track–defines the main length (with its speed depending on what is set on the *global* zone).

This does not reduce freedom, as the tracks can be of arbitrary lengths and the tracks can be not quantized.

The *station* always assumes that the main track is 32[99] beats length.

If the performer creates a very short or very long track, for example one of approximately one quarter of the main track length, this is detected and then the track is played four times per bar (the recognizable lengths are half, one quarter and twice the length of the main track).

98 As of now, *YARMI* does not provide any tool for assigning a sample to a token, so this assignment is done in configuration time, before actually using the instrument. However, we do plan to create some scheme for dynamic assignment, in the hope that it could be used not only for *YARMI* but also for other tangible digital instruments such as the *Reactable*.

99 This is an off-line configuration parameter.

Leader station and inter-station synchronization
As the synchronization between the different *stations* is as important as the synchronization between tracks, we defined that one station is always acting as the *leader station*, and defines when the tracks begin to be played, the performance speed (in BPM), the global volume, etc.

Any station can be the *leader station*. We use a token (the leader token), that, when added to the *global* zone of a *station*, sets it as the leader (the first leader is always the first *station* to connect to the server).

The *leader station* sends its commands to the server, which, in turn, broadcasts them to all the registered *stations*.

Settings
Several configuration parameters can be modified in performance-time. Each setting has an assigned token and its parameter is controlled by rotating it.

Implemented settings are:

- Velocity (byte): a multiplier of the volume set by the leader.
- Quantization (Boolean): specifies whether the sample's positions in a track are snapped to the closest beat.
- BPM (byte): sets the global bits-per-minute.

All of these settings affect only the *station* where they are modified, except in the case of the *leader station* where they affect the whole ensemble.

Visual feedback
Although *YARMI*'s visual feedback is as important as the sound produced, being a project in development, its visual capabilities are in their infancy.

In the current stage the *screen space* shows the table, with the following additions:

- One line for each track.
- One cursor per track showing the current time.
- Several numbers showing the elapsed time, bar count, current main volume and station volume.

Implementation

YARMI's software, like *Puzzling*'s was coded in C++ with OpenFrameworks and ARToolkitPlus; FMOD[100] is also used for audio playback.

Software architecture

As we stated before, *YARMI* implements a client-server architecture, with every *station* identical to each other.

Stations have a GlobalState object that models all the *station*'s information. This object is updated by an independent thread in charge of computer vision and is polled by the sound and video managers (see Figure 24).

Each time a setting is changed in a *station*, it notifies the server, which, if the station is the current *leader* broadcasts the new settings to all the registered *stations* (if it is not the leader the server ignores the new setting, therefore, all the stations act the same, regardless if they are leaders or not).

Conclusions

Although *YARMI's* design is in a mature phase, and in the frame of this thesis we have constructed a working prototype, it is still project in development for which much work is yet to be done.

A major milestone still to be reached is to test *YARMI* in a real performance setup; so far, we have only used it in our lab, in a controlled environment.

We believe that *YARMI* has some characteristics that can turn it into a capable, and interesting, musical instrument

Its networking approach, with many identical components that synchronize themselves automatically, allow for a confident use (delegating some of the cognitive work of the performance onto the system), while maintaining the performers engagement, independence and expressivity, which, in turn are levered with the inclusion of the *immediate zone*.

This combination of the *track* and *immediate* zones offer the “safety” of computer-sequenced music with the expressiveness of traditional musical instruments.

Finally, the explicit representation of the instruments’ feedback, together with the performance happening on a virtual space external to both the audience and the performers allow the public to decode some of the

[100] http://www.fmod.org/

performance aspects, re-installing the lost dimension of virtuosity into the performance.

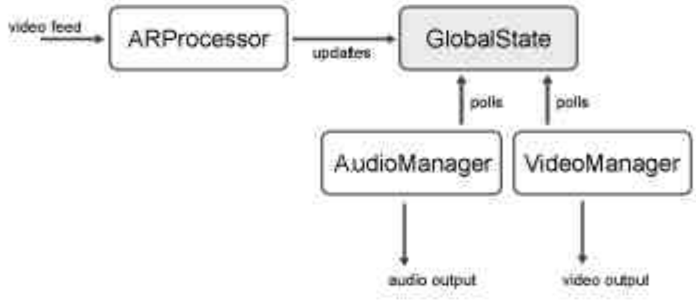

Figure 24 – *Station*'s high-level architecture.

Virtuosity has traditionally played an important role in live musical performances with an added aesthetic dimension of its own. But, for virtuosity to play that role, the audience must know beforehand the instruments' design in order to infer the gestures that produced the sound, or be able to observe and understand the performance's details.

With *YARMI*, once again, the audience can enjoy not only the sonic output of the instruments but also *how* the sounds are created.

Future work
Besides improving the instrument's output (specially its visual output), some paths appear worth to follow.

Specifically, we would like to perform additional research in the following directions:

- Testing and refining. Although we have constructed an encouraging first prototype, the specific goals must be tested for us to be able to refine the proposal. Crucial is to maximize the audience's ability to understand the performance.
- Active inclusion of the audience: if the public can decode the performers' gestures, the next step is to allow them to actively participate in the performance.
- Geographically distributed performances: having an inherently networked instrument would allow us to explore the relevance of proximity and simultaneity in *YARMI*'s performances.
- More multimodal interaction: we would like also to investigate whether new interaction styles and techniques can be added.

Ribbons: a live cinema instrument

Live Cinema

Live cinema is a term recently coined for a long-standing practice[101]: real-time audiovisual performances, which –in its current incarnation– are real-time collaboration between sonic and visual artists [78].

Although the aesthetics of live cinema has been shaped mainly by VJing[102] (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 [78].

This expansion, together with the images of club VJing, has led to mainly produce very abstract and synaesthesia-focused works that somewhat deny traditional cinematographic narrative techniques and methods.

This biases the production, focusing only in "the transitions, the movements, the pure visual beauty" [78]. 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.

---

[101] It indeed is a long-standing practice, since the Wayan Kulit –Indonesian shadow theatre projected by fire– in the XX century; there have been different *projected* real-time performances. Live cinema has its direct predecessors in the Magic Lantern performances of late 1700s and early 1800s, also –of course– in cinema and –perhaps more notably– in the synaesthetic efforts of Color Music and Lumia. For a history of Live Cinema and its artistic language, see [78] and [37].

[102] We refer to VJing as real-time video *mixing* of footage, while live cinema also includes footage creation and its aesthetic, including –but transcending– the appropriation of footage by VJs.

Visual Lutherie and Human-Computer Interaction
These performance-oriented tools that produce moving images are called visual instruments, and therefore, their crafting should be called *visual lutherie*[103].

Paraphrasing Miller Puckette's quote[104] about computer music software we can state that software design cannot help but affect what computer visual production looks like.

As we mentioned in chapter 2, many guidelines and techniques of HCI are applicable (if in consonance with the artist's desires) in new media art production, and can aid in instruments' design [66], including user-centered design, iterative design and direct manipulation.

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.

The Instrument

Design

Traditional cinema *projects* its 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 many times it can be altered by using multiple projection screens that break the traditional rectangle or by using *projection-mapping*[105] techniques.

But 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, on its core, a video player (the user must add video loops to the instrument before the performance) and its able to reproduce videos and live input in a standard way (a full-screen flat representation), and to apply some basic effects, such as transparency, scratching and direct access, its design challenges the flat representation by projecting the cinematic material onto a three-dimensional, virtual, radically deformable canvas.

To be able to do so, *Ribbons* uses a fairly standard particle system: it creates a grid of three-dimensional particles, each one tinted with the

---

[103] Although a rather obvious name, we have never seen it before writing this paper, nor were we able to find any paper or work that uses it.

[104] Again, see *Composed instruments* in chapter 2, *Human-computer interaction*.

[105] That is, projecting over non-flat surfaces, see –for example– White Void's Polygon Playground at http://www.whitevoid.com] or AntiVJ's works at http://antivj.com.

corresponding color of the video. These particles can be manipulated by the performer in novel ways, thus adding a new dimension of expression, orthogonal to the footages' original one and not only distinct from common VJing techniques but also complementary.

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.

The particles can then be used as input for different *visualizations* (such as triangle trips, cubes or lines), which we shall present later.

In the design and construction of the instrument, three axes guided our work: playability vs. autonomy, expressiveness vs. narrative, and originality.

Playability vs. autonomy
The defining characteristic of an instrument is that it is playable. Ideally, users should feel that both the manipulation is as direct as possible, even to the point that it disappears from their cognitive universe as they 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.

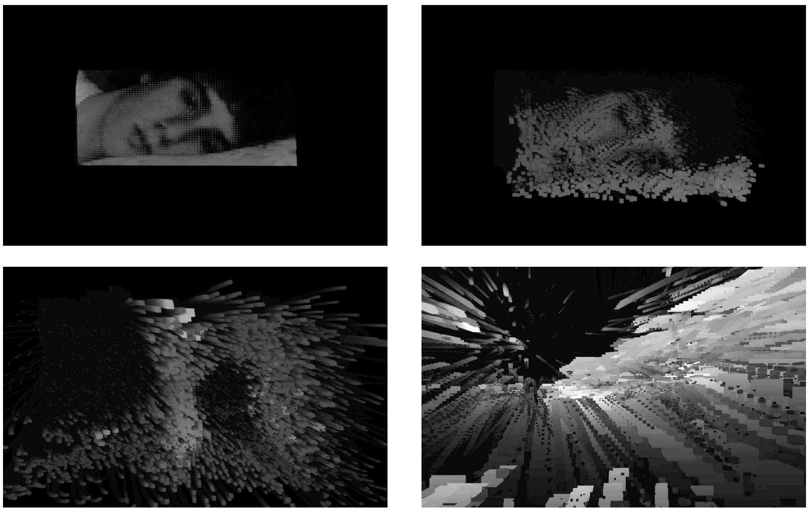

Figure 25 - Different stages of deconstruction of the filmed image by applying a Perlin wind.

In order to reinforce the directness, all the commands built in *Ribbons* trigger an immediate response[106], and the user can directly control

[106] This depends on the hardware being used to run Ribbons. In the tested setup (an Intel MacBook Pro laptop) the achieved performance allowed for immediate response.

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, some times one needs to focus on something else (like 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 (or line in) 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 (see Figure 25), and the particles act as if attached to strings (and then will oscillate and eventually converge to its 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[107]) and the performer can have medium to little control of each particle actual movement, always being able to modify global parameters like the strength of the strings, the direction of the particles, etc.

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

Expressiveness vs. narrative

As we mentioned, the performer can, for example, apply a Perlin "wind"[108] to the particles and deform the projection surface, even to the point of de-constructing the video frame, re-signifying its components, the pixels, as elements capable of independently convey meaning.

---

[107] Perlin noise is a coherent noise function introduced by Ken Perlin. It consists of a very efficient function of either geometrical or geometrical plus temporal variables that uses interpolation between a set of pre-calculated gradient vectors to construct a value that varies pseudo-randomly over one or many of its parameters. See [99] for more information.

[108] A commonly used technique in computer graphics consists on calculating a field of Perlin noise and using it as a vector field of speed for –for example– particles. In these cases, the particles appear to be *blown* by the Perlin noise.

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.

Originality

Our third and last axis of work simply consisted in the attempt of generating a distinct, recognizable visual output (see Figure 26 for some screenshots).

Although we believe that we were moderately successful at it, 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 Alva Noto in his latest tour[109].

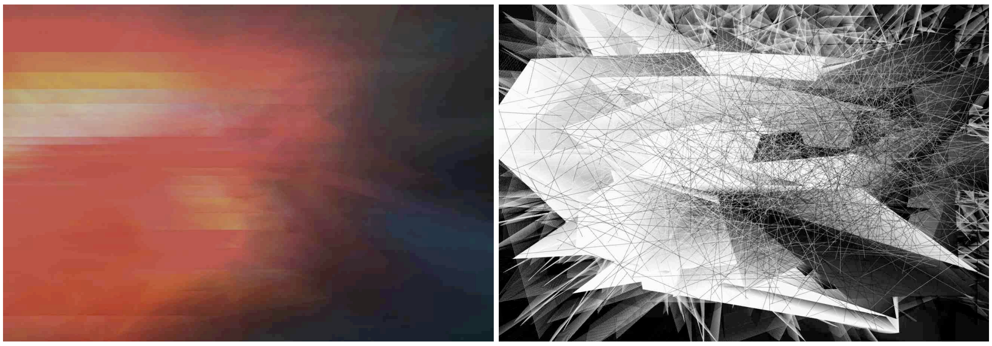

Figure 26 - More Ribbons screenshots: triangles (left) and triangles + lines (right) visualizations.

However, it is obvious that the choices on whether or not use these visualizations or how to combine them is on the performer.

Operation

Ribbons is to be controlled with one hand in the computer's keyboard and the other one in a drawing tablet[110]. There are four different types of commands:

- *Selectors* select a video source or visualization with a keystroke.

[109] UniTXT, see http://www.raster-noton.net/.

[110] Although it can be controlled with a standard computer mouse instead of the drawing tablet, the direct mapping from tablet-coordinates to screen-coordinates allow Ribbons to give an implicit feedback of the current level of the parameter being manipulated and allows the performer to manipulate the instrument with much more precision and speed than what can be achieved using a mouse.

- *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.
- *Control commands* are meta-commands (i.e. not belonging to a Ribbons' performance but commands for settings, quitting, saving, etc.).

In the current version of the instrument the following commands have been implemented:

Selectors
The topmost and bottommost lines of the computer keyboard are destined to selectors. The user can select up to ten videos from ten different banks: by pressing shift plus a number key the user selects a bank and the videos are mapped to the 10 numbers of the keyboard.

The bottommost row keys select the visualizations by turning them on and off with a key press (the used keys, so far, are z for particles, x for triangles, v for video, b for lines and n for UniTXT-like lines).

Tab switches from live video to pre-recorded footage and vice-versa.

Triggers
Triggers and faders use the middle keyboard rows. There are triggers that start, stop or mirror the camera rotation, change the way new frames are drawn (old frames can be erased or faded out), add different levels and directions of Perlin or pure random flow to the particles, draw rectangles or text, that turns on or off some filters, etc.

These commands, together with the faders, are mapped onto the two middle rows of keys.

Faders
As we mentioned, *Ribbons* offer some gestures for the direct manipulation of continuous (real, in [0,1]) parameters. The available gestures are: positioning, moving, dragging, and clicking (touching the tablet with the pen's tip).

All the faders are mapped in an absolute way to either vertical or horizontal displacement. The selection of the parameter is done by holding a key pressed and choosing one of the gestures (direction plus touching or not the tablet with the pen).

The user can control, for example, the transparency of a visualization, the speed of the video playing or the rotation of the camera, or can access a specific point on the video (and scratch by dragging the pen).

Control

By pressing the function keys on the computer's keyboard the user is able to record the performance on the hard disk drive, to edit the camera's parameters, or to turn on or off the unprocessed monitoring of the camera's input.

Implementation

Ribbons is fully implemented in C++ and OpenGL using, again, OpenFrameworks as a programming framework.

Conclusions and future work

We have shown our visual instrument *Ribbons*, which is not only theoretically consistent, but also has been successfully used in "real life" performances (see Figure 27 - *Ribbons'* performing with Uruguayan band Multiplexor. Video, lines, and particles visualizations.), where it provides the performer engagement that is expected from an instrument, while also being able to perform autonomously (even if it is for brief periods, we find particularly interesting this delegation to an automated process, which also means to delegate some actual performative decisions to a previous ourselves).

Figure 27 - *Ribbons'* performing with Uruguayan band Multiplexor[111]. Video, lines, and particles visualizations. Photography by Eugenia Bellini.

[111] http://myspace.com/mltplxr

The instrument allows us to investigate and question the basic need of expressive footage, and its relation with generative visuals and its 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 (that is, not virtual like, for example, projection mapping) solutions.

Future work

We plan to keep on working on *Ribbons* by adding new visualizations, new automated sound-reactive behaviors, and the ability to use simultaneously multiple cameras.

We would also like to work on its performance (in terms of speed), as it can be improved by using computer graphics' techniques.

5 CONCLUSIONS

Technologies often tend to develop faster than the rhetoric evaluating them, and we are still in the process of developing description for arts using digital technology as a medium—in social, economic, aesthetic respects.

Christiane Paul, Digital Art [96]

In this work, we have presented the area of new media art, focusing on its relationship with computer programming and human-computer interaction. We have also presented some of our works that exemplify these relationships and the cross-fertilization between the areas.

This relationship –one of mutual inclusion– between technological and artistic production is defining; new media art not only feeds on technological production, but also reconfigures technology into a *medium* of artistic expression.

Although the relationship between art and technology is as old as art itself, this appropriation of the technological creation is a new phenomenon, called *media appropriation*.

Media appropriation does force new media artists to develop a fair knowledge of their media, definitively blurring the frontiers between artistic, scientific and technological production.

Even taking into account that it is not possible to propose a rigid taxonomy for new media art oeuvres that provides disjoint categories for all the production, it is important to be aware of the formal aspects upon which the art is based.

"Ultimately, every object is about its own materiality, which informs the ways in which it creates meaning" [96], materiality that often takes a known form, such as film, video, and animation; Internet art and software art; and virtual reality and musical or visual environments.

In addition to some systematization of the typology of the production, there are also *themes* that systematically emerge, as they pose natural questions for the new media art realm.

One of these themes, the nature of the interaction between art objects and their public, constitutes an especially central problem on new media art, as interactive artworks constitute the only ones where the very physicality of the object changes because of its consumption.

The interaction, then, becomes not only aesthetically relevant but also defining of the artistic fact.

Consonant with Krueger's definition of the *aesthetics of interaction*, we believe that the cross-fertilization between art and technology can open new horizons, as the field of human-computer interaction has matured.

HCI in interactive new media art can then play a role similar to musical theory, providing artists with a framework from which to construct their artistic language.

This is particularly easy to see in the construction of performative instruments: *digital lutherie*, the creation of musical instruments, and *visual lutherie* (a term here proposed) the creation of tools for visual expression.

As with the rest of new media art, this creation does not have to be digital; yet, the digital medium has an enormous ductility (in the already quoted of García Canclini "everything is possible") that makes it –once appropriated– the natural canvas for new media art production.

If Manovich's principles do characterize new media, it is only when media is appropriated that this artistic ductility emerges, being an almost direct consequence of new media's programmability (Manovich's second principle).

And, as a direct consequence of new media being programmable, automatable, computer programming plays the most defining role in media appropriation.

An attempt to depict computer programming's role is provided by Fishwick's *aesthetic computing* framework, which decomposes the relationship between art and programming in three levels: cultural, implementation and representation.

This framework, although useful, does not take into account some of the different aesthetical appropriations of computer programming itself, failing, for example, in the depiction of the visual programming within the arts.

An interesting aspect of visual programming is that it instantiates one of the most important techniques of human-computer interaction: rapid prototyping.

In effect, art-oriented visual programming languages provide a great amount of pre-coded objects that are offered to the programmer. These objects are not only useful because they abstract artists from some of the technical difficulties of, for example, data processing, but also because they allow for gestalt-like problem-solution heuristics.

By having a direct access to a direct representation of these objects, artists can translate solutions into the metaphorical world proposed by the visual language.

Aesthetic computing does offer a formalization of the aesthetic relevance of, for example, Cabezas' *C5*, which shows a very direct pairing in the representation level.

On the other hand, aesthetic computing does not suffice to describe more extreme works such as Morgan's *Piet*.

While *C5*'s structural pairing was intense (although it may appear too indirect at first), in *Piet* it is nonexistent.

Indeed, if at the representation level we are to see the relationship between the code's morphology and its behavioral output, being *Piet* designed so that its source code resembles Piet Mondrian's paintings, it does not, a priori, show any kind of structural pairing.

Piet does pose an extreme view of computer programming within art as it reformulates a tool –programming– as an art object, constituting a wonderful example of media appropriation, almost negating programming language traditional role.

Both *Piet* and *C5* have a very rich cultural layer, being both inspired by traditional painters' works. It is probably not a coincidence the many similitudes between Torres García *Constructivismo Universal* and Mondrian's neoplasticism (*The Stijl,* the style), or their constructivist roots.

The artworks

The theoretical framework is very present in the three pieces shown here: *Ribbons*, *YARMI* and *Puzzling*.

In particular, the influence of computer engineering in their realization was very relevant. Beyond the obvious importance of software literacy, human-computer interaction techniques were used in both the development of the interaction and the evaluation of its aesthetical proposal.

Ribbons' interaction scheme, for example, is based on a drawing tablet that offers implicit feedback on the values of the variables being controlled while providing the means for a direct manipulation based interaction.

The performative aspects of *Puzzling*, which, as an installation, is aimed towards casual users on a gallery space, have been tested by using early prototyping and iterative design: two very important techniques of interaction design.

This concept of the public of an art object being users of a performance-based installation, where the art fact emerges from the interaction, can be reductionist to some extent; however, it offers an extremely valuable insight on *how* to develop such interaction to obtain the desired interaction.

Obtaining the desired interaction equals to obtaining the desired art object, as it is *in* the interaction where the interactive art object exists.

Also interesting is the pairing between the programmatic gestures, in a very refined technological language, and the graphical results on *Ribbons*.

Creating a new media art piece requires to code into the technological language the desired aesthetic result. This codification can be achieved by computer programming, as in *Ribbons,* or it can be lower level manipulation as in the *Tecnocordio*.

The third piece presented, *YARMI*, constitutes the most ambitious project: it creates a musical and visual instrument that aims to be able to be played by anyone while, at the same time, providing an environment for very refined musical expression.

This easiness of use is based in using well-known metaphors and interaction techniques and in direct manipulation (actions that trigger immediate and distinct responses are easier to understand and remember).

The explicit representation used for tracks, samples and time are standard and do not require any musical knowledge (different from, for example, *Reactable*'s interaction based on low frequency oscillators).

It also aims to recover the dimension of virtuosity within digital lutherie: musical performances whose gestures can be decoded by the audience.

Thanks to that decoding (which, in *YARMI*, is based on shared augmented reality space between performers and public, on the direct relationship between interface and sound that sequencers provide, and on the utilization of standard interaction artifacts), an almost forgotten aesthetic dimension is recovered: the enjoyment of *how* the music is constructed within the interaction scheme proposed by the instruments used.

In addition, the aforementioned *cognitive boundary* between virtual and actual data representation, or between digital and physical interaction is addressed and plays a defining role in *YARMI* as its tangible interaction is explicitly presented to the audience, shaping the artistic performance.

YARMI is still a work in progress. Within this thesis we have constructed a working proof-of-concept and concluded its design, both as an instrument and as a piece of software.

Geographic specificity

A final conclusion of this work is about the necessity of developing a geographic specific (new media) artistic language.

If, as Ludwig Wittgestein claimed, “the limits of our language are the limits of our world, in describing the world, we create it”, the artists' creation cannot be based on other peoples' words.

For artists in peripheral countries it is a main necessity to develop location specific artistic languages that allow for media appropriation schemes that respond to local artistic needs.

If technology is taken as something given (not unchangeable but with a frame of work already defined), some of the aesthetical parameters that shape art production obey to foreign interests.

Against this, we, as peripheral artists, scientists, and technologists, shall rebel, by finding our own, new, paths for art expression. For media to be appropriated knowledge is needed but, as Eladio Dieste[112]'s quotes showed, a change of attitude is also indispensable.

[112] See *Digital inclusion* in chapter 1, *New media art*.

6 REFERENCES

[1] Abras, C., Meloney-Krichmar, D., and Preece, J. 2004 User-Centerd Design. In Encyclopedia of Human-Computer Interaction, W. Bainbridge,Ed Thousand Oaks: Sage Publications.

[2] Accot, J. and Zhai, S. 1999. Performance evaluation of input devices in trajectory-based tasks: an application of the steering law. SIGCHI conference on Human factors in Computing Systems.

[3] Adamczyk, P. D., Hamilton, K., Twidale, M. B., and Bailey, B. P. 2007. HCI and new media arts: methodology and evaluation. Workshop held as part of CHI 2007.

[4] Ahmad, R. 1999. Visual languages: a new way of programming. Malaysian Journal of Computer Science. 12, 1, 76-81.

[5] Apple 2008. Quartz Composer.

[6] 2006 Support for the Arts Handbook. Australian Goverment.

[7] Azcona, M. M. 2005. Making Sense of a Multi-Protagonist Film: Audience Response Research and Robert Altman's Short Cuts (1993). Miscelánea: A Journal of English and American Studies. 32, 11-22.

[8] Azuma, R. T. 1997. A survey of augmented reality. Presence-Teleoperators and Virtual Environments. 6, 4, 355-385.

[9] Baars, B. J. 1993 A Cognitive Theory of Consciousness. Cambridge University Press.

[10] Balcom, D. 1996. Short Cuts, Narrative Film and Hypertext.

[11] Barthes, R. 1967 Elements of Semiology. Hill and Wang.

[12] Barwood, H. 2000. Cutting to the chase: Cinematic construction for gamers. At Game Developers Conference. 2.

[13] Baumer, E. and Tomlinson, B. 2007. Advocating Polytheoretic Evaluation of Interactive Art and New Media. Workshop held as part of CHI 2007.

[14] Bertelsen, O. W. 2002. Aesthetics as Means for Supporting Development in Use - Beyond the Designed Purposefulness. Aesthetic Computing. Dagstuhl-Seminar-Report.

[15] Boshernitsan, M. and Downes, M. 1997. Visual programming languages: A survey. Technical Report UCB/CSD-04-1368. University of California, Berkeley,.

[16] Bramann, J. K. 1998. Understanding the end of art.

[17] Institute, B. F. 2006. Critics' Top Ten Poll 2002.

[18] Burnett, M. M., Baker, M. J., Bohus, C., Carlson, P., Yang, S., and van Zee, P. 1987. Scaling Up Visual Languages. IEEE Computer. 28, 3, 45–54.

[19] Burnett, M. M., Baker, M. J., Bohus, C., Carlson, P., Yang, S., and van Zee, P. 1995. Scaling Up Visual Programming Languages. Computer. 45-54.

[20] Cabezas, J. J. 2007. The C5 programming language. Technical Report INCO RT 07-05.

[21] Card, S. K., Moran, T. P., and Newell, A. 1983 The psychology of human-computer interaction. CRC Press.

[22] Chadabe, J. 2002. The limitations of mapping as a structural descriptive in electronic instruments. New Interfaces for Musical Expression. 1-5.

[23] Coughlan, T. and Johnson, P. 2007. Support for Creativity in Musical Collaboration and Creator/Technologist Interaction. Workshop held as part of CHI 2007.

[24] Danto, A. C. 1964. The Artworld. The Journal of Philosophy. 61-19.

[25] Danto, A. C. 2004 Art criticism after the end of art. In Unnatural Wonders: Essays from the Gap Between Art and Life, Columbia University Press.

[26] deLahunta, S., Higgins, P., Ishii, H., Moriyama, T., and Ng, E. 2004 Rearview Mirror: 1990-2004. In Cyberarts 2004, H. Leopoldseder, C. Schöpf, and G. Stocker, Eds. Hatje Cantz Publishers.

[27] Dickie, G. 1974 Art and the aesthetic: an institutional analysis. Cornell University Press.

[28] Diehl, S. 2002. Software Visualization. Aesthetic Computing. Dagstuhl-Seminar-Report.

[29] Dieste, E. 1998 Introducción a la teoría. Facultad de Arquitectura, Universidad de la República.

[30] Donovan, A., Miller, S., and Lally, E. 2006. New Media Arts Scoping Study. Report to the Australia Council for the Arts.

[31] Dragicevic, P., Ramos, G., Bibliowitcz, J., Nowrouzezahrai, D., Balakrishnan, R., and Singh, K. 2008. Video browsing by direct manipulation. CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems. 237–246. DOI=http://doi.acm.org/10.1145/1357054.1357096.

[32] Drucker, J. 2006 Interactive, Algorithmic, Networked: Aesthetics of New Media Art. In At a distance: Precursors to Art

and Activism on the Internet, A. Chandler and N. Neumark,Eds The MIT Press.

[33] Duchamp, M. 1957. The creative act. Session on the Creative Act, Convention of the American Federation of Arts.

[34] Duignan, M., Noble, J., and Biddle, R. 2005. A Taxonomy of Sequencer User-Interfaces. International Computer Music Conference, ICMC 2005.

[35] Goodman, E. 2008. Sketching in Hardware Day 1: Tom Igoe.

[36] 2008. Piet - Esolang.

[37] 2006 VJ: Audio-Visual Art and VJ Culture. Lawrence King Publishing.

[38] Fernandes, G. and Holmes, C. 2002. Applying HCI to music-related hardware. SIGCHI Conference on Human Factors in Computing Systems.

[39] Fincher, S. 2008. The Pattern Gallery.

[40] Fischer, H. Crisis of the contemporary art? Which crisis?

[41] Fishwick, P. 2003. Aesthetic computing manifesto. Leonardo. 36, 255-256.

[42] Fishwick, P. 2007 Aesthetic Computing: A Brief Tutorial. In Visual Languages for Interactive Computing: Definitions and Formalizations, F. Ferri,Ed

[43] Flusser, V. 2000 Towards a Philosophy of Photography. Reaktion Books.

[44] Franke, H. W. 1985 Computer graphics – computer art. Springer-Verlag New York, Inc.

[45] Frieling, R. 2004. Form Follows Format, Tensions, Museums, Media Technology, and Media Art. Media Art Net.

[46] García Canclini, N. 1999. La globalización imaginada. Paidós.

[47] Greenberg, I. 2007 Processing: Creative Coding and Computational Art. friends of ED.

[48] Gunderson, P. A. 2004. Danger Mouse's Grey Album, Mash-Ups, and the Age of Composition. Postmodern Culture. 15, 1.

[49] Heim, M. 1993 The Metaphysics of Virtual Reality. Oxford University Press.

[50] Hernsberger, E. 2006. Photography as readymade art. Masters.

[51] Hewett, Baecker, Card, Carey, Gasen, Mantei, Perlman, Strong, Verplank 2008. ACM SIGCHI Curricula for Human-Computer Interaction.

[52] HFES 2009. Human Factors and Ergonomics Society.

[53] Hornbostel, E. M. and Sachs, C. 1914 Systematik der Musikinstrumente: Ein Versuch.

[54] Huhtamo, E. 2004. Trouble at the Interface, or the Identity Crisis of Interactive Art. Media Art History.

[55] Hunt, A. and Thomas, D. 2001. The Art in Computer Programming. Java and Object Oriented Conference.

[56] Jennings, P., Giaccardi, E., and Wesolkowska, M. 2006. About face interface: creative engagement in the new media arts and HCI. Conference on Human Factors in Computing Systems. 1663-1666.

[57] Jordà, S. 2001. New musical interfaces and new music-making paradigms. New Interfaces for Musical Expression, NIME 2001.

[58] Jordà, S. 2004. Instruments and Players: Some Thoughts on Digital Lutherie. Journal of New Music Research.

[59] Jordà, S. 2005. Digital Lutherie: Crafting musical computers for new musics performance and improvisation. PhD.

[60] Kaltenbrunner, M., Jorda, S., Geiger, G., and Alonso, M. 2006. The reacTable*: A Collaborative Musical Instrument. "Tangible Interaction in Collaborative Environments" (TICE), at the 15th International IEEE Workshops on Enabling Technologies (WETICE 2006).

[61] Kay, A. 1987. Alan Kay: Doing with Images Makes Symbols. 1.

[62] Kelleher, C. and Pausch, R. 2005. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. ACM Comput. Surv. 37, 2, 83-137.

[63] Kern, M. 2008. America Platina: Modernity and Its Utopias. Colóquio Internacional História e(m) Movimento.

[64] Knuth, D. E. 1974. Computer programming as an art. Commun. ACM. 17, 12, 667-673.

[65] Koonts, L. 1994. Multi-media conversation between theoretical physics and graphic design.

[66] Laurenzo, T. 2008. HCI in new media art practices. IX Congreso Internacional de Interacción Persona Ordenador 2008, Interacción 2008.

[67] Laurenzo, T. 2008. Low-cost digital lutherie. International Symposium on Electronic Art, ISEA2008.

[68] Laurenzo, T. and Castro, J. F. 2003. Arte y Tecnología. INCO. Bachelor Hons.

[69] Pennsylvania, L. o. t. U. o. 2005. Appropriation Art and Copyright.

[70] Lieberman, H. 2008 A Creative Programming Environment, Remixed. In T. Ericson and D. MacDonald,Eds MIT Press.

[71] Littler, J. 2005. Art and Computer Programming.

[72] Machover, T. 2002. Instruments, interactivity, and inevitability. New Interfaces for Musical Expression, NIME2002. 1-1.

[73] MacIntyre, B., Bolter, J. D., Vaughan, J., Hannigan, B., Moreno, E., Haas, M., and Gandy, M. 2002. Three angry men: dramatizing point-of-view using augmented reality. International Conference on Computer Graphics and Interactive Techniques, ACM SIGGRAPH 2002 conference abstracts and applications. 268-268.

[74] MacKenzie, I. S. 1992. Movement time prediction in human-computer interfaces. Conference on Graphics interface '92. 140–150.

[75] Maeda, J. 1999 Design By Numbers. The MIT Press. Cambridge, MA.

[76] Maeda, J. 2003. Design by Numbers.

[77] Maeda, J. 2004 Creative code: aesthetics and computation from the MIT Media Lab. Thames & Hudson.

[78] Makela, M. 2006. Live Cinema: language and elements. Media Lab. MA in New Media.

[79] Malina, R. 2006 Foreword. In At a Distance: Precursors to Art and Activism on the Internet, A. Chandler and N. Neumark,Eds The MIT Press.

[80] Manovich, L. 2001 The Language of New Media. The MIT Press.

[81] Manovich, L. 2002. New Media from Borges to HTML. The New Media Reader.

[82] McLuhan, M. 1964 Understanding media: the extension of man. McGraw-Hill.

[83] Merriam–Webster 2009. Merriam-Webster Dictionary.

[84] Mitchell, B. 1999. Myron W. Krueger.

[85] Mitchell, W. J. T. 1995 Picture theory: Essays on visual and verbal representation. University of Chicago Press.

[86] Moreno, E., MacIntyre, B., and Bolter, J. D. 2001. Alice's Adventure's in New Media: An Exploration of Interactive Narratives in Augmented Reality. cast01, Living in Mixed Realities. Conference on artistic, cultural and scientific aspects of experimental media spaces.

[87] Morgan-Mar, D. 2008. Piet.

[88] Myers, B. A. 1986. Visual programming, programming by example, and program visualization: a taxonomy. Proceedings of the SIGCHI conference on Human factors in computing systems. 59-66.

[89] Naimark, M. 2003. Truth, Beauty, Freedom, and Money: Technology-Based Art and the Dynamics of Sustainability. J Leonardo/ISAST report. http://www. artslab. net.

[90] Nake, F. and Grabowski, S. 2002. Aesthetics and Algorithmics. Aesthetic Computing. Dagstuhl-Seminar-Report.

[91] Nezvanova, N. 2000. Cycling'74 as quoted by Netochka Nezvanova (archive).

[92] Newton-Dunn, H., Nakano, H., and Gibson, J. 2003. Block jam: a tangible interface for interactive music. New Interfaces for Musical Expression (NIME 2003). 170-177.

[93] Norman, D. A. 1988. The Psychology of everyday things.

[94] Oliver, J. 2008. levelhead.

[95] Patten, J., Recht, B., and Ishii, H. 2002. Audiopad: a tag-based interface for musical performance. New Interfaces for Musical Expression (NIME 2002). 1-6.

[96] Paul, C. 2008 Digital art. Thames & Hudson.

[97] Paulos, E. 2007. HCI Cannot Be Used To Evaluate Art. Workshop held as part of CHI 2007.

[98] Peppler, K. and Kafai, Y. 2005. Creative Coding: Programming for Personal Expression.

[99] Perlin, K. 2002 In the beginning: The Pixel Stream Editor. In SIGGRAPH 2002 Course 36 Notes. Real Time Shading, S. 2002,Ed

[100] Pia, T. 2008 Enactive Cinema. Simulatorium Eisensteinense. Taideteollinen korkeakoulu.

[101] Pinhanez, C. S., Davis, J. W., Intille, S., Johnson, M. P., Wilson, A. D., Bobick, A. F., and Blumberg, B. 2000. Physically

interactive story environments. IBM Systems Journal. 39, 3, 438-455.

[102] University, P. 2009. WordNet.

[103] Processing.org 2008. Processing website, about page.

[104] Puckette, M. 1988. The patcher. International Computer Music Conference. 420-429.

[105] Puckette, M. 1996. Pure data: another integrated computer music environment. Proceedings of the Second Intercollege Computer Music Concerts. 37–41.

[106] Puckette, M. 2002. Max at Seventeen. Computer Music Journal. 26, 4, 31-43.

[107] Quarta, M. 2003. ISS Cube.

[108] Queirolo, P. 1997. Typed Windows: an implementation of a programming language for graphics design. Master.

[109] Rhodes, G. A. 2008. Augmented Reality in Art: Aesthetics and Material for Expression.

[110] Rhodes, G. A. 2005. Metonymy in the moving image: Multichannel cinema. Master.

[111] Rhodes, G. A. 2008. 52 Card Psycho - Re-Imagining Cinema. International Symposium on Electronic Art, ISEA2008.

[112] Rieser, M. 2002 The art of interactivity: Interactive installation from gallery to street. In Computers and Art, S. Mealing,Ed Intellect Books.

[113] Rokeby, D. 1998 The construction of experience: Interface as content. In Digital Illusion: Entertaining the future with high technology, C. Dodsworth,Ed Addison-Wesley Publishing Company.

[114] Rush, M. 2005 New media in art. Thames and Hudson.

[115] Saltzman, M. A. 2000 Game Design: Secret of the Sages. Brady.

[116] Schmit, K. 2006. Dissecting the discourse.

[117] Schnell, N. and Battier, M. 2002. Introducing composed instruments, technical and musicological implications. New Interfaces for Musical Expression, NIME 2002.

[118] Schoch, T. 2006. Obfuscated programming - Piet.

[119] Scrivener, A. 1994. The Impact of Visual Programming in Medical Research. Medicine Meets Virtual Reality II --

Interactive Technology and Healthcare: Visionary Applications for Simulation, Visualization and Robotics.

[120] Serra, X. 2008. Technological Innovation in the Current Social Context: Who is really in control? New Interfaces for Musical Expression (NIME 2008).

[121] Stephanidis, C., Akoumianakis, D., Sfyrakis, M., and Paramythis, A. 1998. Universal accessibility in HCI: Process-oriented design guidelines and tool requirements. Proceedings of the 4th ERCIM Workshop on User Interfaces for All. 19–21.

[122] STMicroelectronics 2002. Seven Darlington Arrays.

[123] Tanaka, A. and Knapp, R. B. 2002. Multimodal interaction in music using the Electromyogram and relative position sensing. New Interfaces for Musical Expression. 1-6.

[124] Intruments, T. 1988. 7486, Datasheet.

[125] Tidwell, J. 2009. Designing Interfaces (web site of the homonymous book).

[126] Treglown, M. 1994. Qualitative models of user interfaces. Proceedings of the conference on People and computers IX table of contents. 261-272.

[127] Tribe, M., Jana, R., and Grosenick, U. 2006 New Media Art. Taschen.

[128] Velmans, P. M. 1998 Physical, Psychological and Virtual Realities. In The Virtual Embodied, D. Wood,Ed Routledge.

[129] Vertegaal, R. and Eaglestone, B. 1996. Comparison of input devices in an ISEE direct timbre manipulation task. Interacting with Computers. 8, 13-30(18). DOI=doi:10.1016/0953-5438(95)01015-7.

[130] VVVV.org 2009. VVVV online documentation.

[131] Wallraven, C., Kaulard, K., Kürner, C., Pepperell, R., and Bulthoff, H. H. 2007. Psychophysics for perception of (in)determinate art. APGV '07: Proceedings of the 4th symposium on Applied perception in graphics and visualization. 115–122.

[132] Wanderley, M. M. 1999. Non-obvious Performer Gestures in Instrumental Music. GW '99: International Gesture Workshop on Gesture-Based Communication in Human-Computer Interaction. 37–48.

[133] Wanderley, M. M. and Orio, N. 2002. Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI. Computer Music Journal.

[134] Ware, C. 2004 Information Visualization: Perception for Design, 2nd. edition. Morgan Kaufmann.

[135] Watzlawick, P., Bavelas, J. B., and Jackson, D. D. 1967 Pragmatics of human communication. W. W. Norton & Company.

[136] Weinbren, G. 1997. The Digital Revolution is a Revolution of Random Access.

[137] 2009. Max (software).

[138] Wilson, G. 2006. Off With Their HUDs!: Rethinking the Heads-Up Display in Console Game Design.

[139] Wise, H. 1973. At the Leading Edge of Art. Electronic Arts Intermix. 8.

[140] Wollheim, R. 1980 Art and Its Objects. Cambridge University Press.

7 INDEX OF FIGURES

8 PUBLISHED PAPERS AND PERFORMANCES

This appendix shows the papers that this work permitted to publish, and the performances and exhibitions that its products also permitted.

Papers

T. Laurenzo, E. Rodríguez, J. Castro. *YARMI: an augmented reality musical instrument*. New Interfaces for Musical Expression, NIME'09, Philadeplhia, USA.

T. Laurenzo, *Ribbons: a live cinema instrument*. International Symposium on Electronic Art, ISEA2009, Belfast, UK.

T. Laurenzo, E. Rodríguez. *Puzzling*. IV Symposium Iberoamericano de Computación Gráfica, SIACG2009. Isla Margarita, Venezuela.

T. Laurenzo, *HCI practices in new media art*. IX Congreso Internacional de Interacción Persona – Ordenador. Interacción 2008, Albacete, España.

T. Laurenzo, *Low cost digital lutherie*. International Symposium on Electronic Art, ISEA 2008, Singapur, Singapur.

T. Laurenzo, *Arte con nuevos medios e inclusión digital en Uruguay*. Congreso internacional de arte y nuevas tecnologías, organizado por el Museo de Arte Contemporáneo de San Pablo, Brasil, 2007.

Performances

Live cinema with Uruguayan band MUX:

Ciclo *Movimiento*. Montevideo, Uruguay.

Museos en la Noche. Montevideo, Uruguay.

Live cinema performance commissioned by Uruguay's design cluster. Montevideo, Uruguay.

Presentation and performance at *Dorkbot.MVD*. Subte Municipal Gallery. Montevideo, Uruguay.

Short-listed for performance at the International Symposium of Electronic Art, ISEA 2009. Belfast, Ireland.