Soundpainting language recognition

DH Master thesis

By Arthur Parmentier

Under supervision of Sarah Kenderdine at EM+

I am grateful to several people that have made my work possible and helped me during my research. In particular, I want to thank my supervisor Sarah Kenderdine of the EM+ laboratory of EPFL for supporting my project and offering me the resources to achieve it, Lorenzo Cantelli for his technical support and advice since its early days and the whole EM+ team for their welcome and attention. I would like to thank again Ronan Boulic and Ken Déguernel for the time and assistance they have dedicated to me during all phases of my work and express all my gratitude to Constance Frei for her attentiveness and previous help from the early conceptualization to the final polish of this report. I also thank Patrick Donaldson, Jean-Cédric Chappelier, Mathias Delahaye, Patrick Chouard and the ARTLab team of EPFL for their support.

I am also thankful to the many people that I met long before my master thesis and without whom I would not have the chance to present this work now. I am first thinking about my parents who gave me care and support during these long years at EPFL and my sisters who I grew up with. “Ce soir est un grand soir” and I give my most sparkling and passionate thanks to Loni Mahé, Valentin Prost, Eric Willimann, Albane Descombes, Anne-Michèle Savoy, Simon Prêcheur Llarena, Martin Sochor and Xavier Suermondt for the musical and soundpainting adventures we shared these last years, including the whole Musical - AGEPoly team and people who participated in creating such an inspiring artistic world at EPFL. Finally, I am deeply touched by the many people who have hosted and welcomed me in recent years, in particular within the AJJF association in Lausanne.

My most poetic thoughts are for the strangers, artists, teachers, soundpainters from all over the world who opened to me some of their universe, made me grow and accompanied me in paths I haven’t crossed before. I would like to mention here Cyrille Colombier, Fanny Moussey, Ana Candida Machado, Taiyo Omura, João Paulo Prazeres, Edite Rocha, Mario Marques Trilha, Victor Mendes, Jennifer Candeias, Emi Ayò, os grupos de Soundpainting em Rio (Cineorquestra e Baila Baila Big Bando) e Belo Horizonte, the association Zellidja, Walter Thompson, Marie Monfrais, Johanna Gonçalvez Martin and Olivier Thouault.

Table of contents

Introduction

This master thesis is the result of 17 weeks of research on computer recognition and on the linguistics of Soundpainting. Soundpainting is a sign language developed by the New York composer and saxophonist Walter Thompson from 1974 for real-time composition with his orchestra. Although the language was originally used for composing with musicians, it has extended to multiple artistic disciplines such as dance, theater of visual arts and is now used worldwide by a variety of artists in diverse contexts and configurations. For a short demonstration of a typical Soundpainting performance by Thompson, see this video. Additional explanation of signs and gestures in Soundpainting are given in Thompson *Workbooks* 1 (2006), 2 (2009) and 3 (2014).

Soundpainting is originally not a language designed for working with electronic devices and computers. It is often reported that their use in Soundpainting is made difficult by the high reactivity requested by the soundpainter, usually the composer, to the set of performers that forms the orchestra. However, digital tools have been used all along the XXth century in new forms of compositional processes and esthetics of music. Today, most music production makes use of computers and their technologies, while the development of computer-aided tools led to the emergence of "computer music" as a genre in its own right. Moreover, they have an increasing potential for learning, performing in real-time and exploring new artistic materials with creativity.

In the theoretical part of this report, I first contextualize Soundpainting in the broader movements of the XXth centuries such as the search for new languages, the abandon of tonality, the creation of new timbres and the mosaic of music genres that emerge in this epoch. Then, I show that the use of signs in music has a long history and how Soundpainting builds upon latent representations of sound that have developed in centuries. Recent theories of signs and common descriptions of Soundpainting as a language lead me to a linguistic approach, which its current literature only covers superficially. Inspired by French structuralism, I investigate its language mechanisms at several levels, from the creation of a sign to some of its basic grammatical production rules.

In the practical part, I present a sign recognition program which allows artists to control virtual instruments or devices using gestures. In particular, this tool has been designed to identify simple Soundpainting gestures and implements its grammar in order to form sentences, i.e. complex requests out of a temporal sequence of signs. However, it can also be used with a custom set of signs, that the user can define and train himself in the program, in analogy with the developments of Soundpainting during which artists have created new signs for concepts from several artistic disciplines or everyday life objects and actions. The program itself is created with Max/MSP, a visual programming language for music and multimedia created at IRCAM in 1985. It is a popular choice among interactive multimedia artists and I was able to learn it quickly in the early stages of my master thesis. Additionally to the recognition of signs that I propose, it has an important potential for combining computer-aided composition, sound and video synthesis or interactive interfaces for the future.

Both parts communicate with each other and are complementary. While the theoretical part precedes the practical part in my report, important conclusions and derivations that I present in it are inspired by my construction of the recognition tool. On the other hand, the theoretical part explains some of my implementation choices and motivates the structure of the tool in terms of several independent layers, each performing a specific function. Unlike recognition tools for languages that are already well described, my research is a back and forth between understanding the implications of linguistics on the choice of specific recognition tools and analyzing what my implementation choices, and practical observations unveil about the theoretical structure of Soundpainting itself.

In the writing process and the building of my recognition tool, I am mainly targeting the Soundpainting and Max/MSP communities that I consider myself part of. I am in that sense both an observer and an actor in all the points described and discussed in the theoretical part. A certain level of experience and familiarity with the concepts introduced and developed in this report is excepted on both Soundpainting and Max/MSP programming. Although I have tried to clearly present and define most of the key elements of the discussion, the reader is invited to refer himself to the bibliography and to my additional notes when necessary.

My research and proposal are oriented by several influences. First, I would like to mention my initial background in physics (and surrounding fields such as mathematics) at EPFL, which introduced me to an important set of scientific models and general scientific analytic tools. Then, I have followed a masters in the young field of Digital Humanities, interested in studying, representing and questioning human practices and interactions with digital tools. It has given me an overview of machine learning, of user experience and several critical approaches to the digital world in general. Aside from the academic world, I am a musician (percussionist), former co-leader of the EPFL Soundpainting group and see myself as a traveler, interested in anthropology, cognitive sciences and linguistics, among other topics. In recent years, I have paid an increasing attention to the French structuralist approaches in anthropology, linguistics and social sciences in the works of Levi-Strauss, Foucault and Deleuze. But aside my theoretical readings, I had the opportunity to realize two field works on the topics of art and digital humanities: one in Brazil, 2018, where I had the chance and honor to meet and perform with the Soundpainting groups of Rio de Janeiro and Belo Horizonte; a second one at the triple frontier between Brazil, Colombia and Peru in 2019, where I could work in an interdisciplinary project about the use of digital tools in indigenous communities and language "preservation" in the XXIth century. Before starting this project, I have been in touch with Philippe Spiesser (HEM Geneva) and Pierre Donat-Bouillud (IRCAM) about they work in GeKiPe, a research project on sound-movement interaction focusing on sound percussive sound synthesis and control from iconic gestures captured by motion tracking gloves. Finally, after a first invitation of Thompson at EPFL in 2018 and a "mountain Soundpainting" workshop with Marie Monfrais in 2019, my master thesis started with the second invitation of Thompson at EPFL for a one-week multidisciplinary workshop concluded by an improvised soundtrack on 3 silent movies.

I. Theoretical part

A. A brief history of Soundpainting

1. Back in Woodstock 1974: emergence in emergency

The emergence of Soundpainting (thereafter SP) is well documented by its creator Walter Thompson in the first SP workbook *Soundpainting, The Art of Live Composition* (Walter Thompson, 2006, pages 12-13), in which he explains that using gestural signs came up as an emergency response to the loss of control of his orchestra during a performance:

> Woodstock during the 1970s was an important place for the growth of creative music. This was largely due to The Creative Music School (CMS), founded by Karl Berger whose vision was to invite composers and performers such as John Cage, Ed Blackwell, Carlos Santana, Don Cherry, Anthony Braxton, and Carla Bley among others, to give 2-week master classes with the students and often each session would culminate with a performance. […] The CMS was, in the earlier years, closed during summer and many of the students would stay and live in Woodstock until the school re-opened. During Thompson's1 first summer he organized jam sessions with these students and formed his first orchestra. His group included 22 musicians and 7 dancers – the dancers would improvise to material performed by the musicians. Thompson produced a series of 3 concerts at Woodstock's Kleinert/James Gallery. The focus of his work with the orchestra was on compositions incorporating sections of open form improvisation. It was during these early days that Thompson created his first signs that would later grow to become the Soundpainting language. The first gestures of Soundpainting were created in the moment during the opening concert. Thompson had notated a composition where the basic rule for each player, when performing a solo improvisation, was to make a relationship to the notated material. The first person to take a solo did not follow the rule and Thompson felt that in order to maintain the integrity of the piece he needed to come up with a way to guide the composition behind the soloist. He decided to create a sign asking the other performers to play a specific content behind the soloist. In the moment, he created the gesture Long Tone and pointed at several players and gestured to them to begin. The Long Tone gesture was easily understood and executed by the

[1] Here, Thompson writes about himself in the 3rd person; he is the author of the text.

> performers and a few minutes later Thompson created the gesture Pointillism which was also readily understood and performed. After the concert Thompson continued to develop his signs and over the next few years in Woodstock Thompson would develop 40 new gestures.
>
> In 1980 he moved to New York City and formed The Walter Thompson Orchestra (then known as The Walter Thompson Big Band). During the first year with his orchestra, while conducting one of his notated works in concert in Brooklyn, New York, Thompson wanted to communicate with his orchestra and decided to use some of his signs. Trumpet 2 was soloing and during his solo Thompson asked one of the other trumpet players to create a repetitive background behind the soloist and he signed the phrase: Trumpet 1, Background, With, 2-Measure, Feel; Watch Me, and then beat a 4 pattern to bring Trumpet 1 in but, Trumpet 1 did not respond, he stared at Thompson with a blank look. Thompson had never used his signs with his new orchestra and they had no idea what was meant by the gestures. A week later at rehearsal several members of the orchestra asked what the signing was about. Thompson taught the orchestra a few signs and they responded favorably to it and encouraged Thompson to continue developing the language. During the next 10 years, Thompson developed Soundpainting into a comprehensive sign language comprising more than 200 gestures for composing in real time with musicians.

Thus, Soundpainting has emerged in emergency. At the opposite of oral and written languages that have developed in thousands of years, Thompson had to construct signs and a syntax that he thought the orchestra would understand on the fly without being able to discuss prior conventions. Even though later, the signs were discussed and learned with conventions, the fact that they must be quickly understood had important consequences (see I.A.2.b)(2)).

On top of the creation of his first signs, Thompson describes the most common configuration of SP: a single composer (sometimes a few) faces an orchestra of performers. The soundpainter creates requests to the performers using the sign language in real-time, during the performance. At the exception of two signs, the performers do not sign to the composer but rather propose material that he will shape and compose with.

2. Developments

a) A multidisciplinary language

One of the most important developments of SP in the 90's and later is its use with several disciplines: music, dance, acting, all visual arts, sculpture… Recently, SP was even used to control a swarm of drones (Couture, Bottecchia, Chaumette, Cecconello, & Rekalde, 2017), showing the diversity of instruments and disciplines it can be meaningful to. Thompson explains that:

> Performers first learning Soundpainting won't relate it to any specific discipline because of its multidisciplinary applications. [He] created many of the gestures from concepts found in theatre, music, dancing, visual arts, and happenings in everyday life. Since Soundpainting is not a discipline-based language dancers, musicians, actors, and visual artists never have difficulty understanding the meaning of the gestures.

To the question "Are all your gestures intended to be interpretable by both musicians and dancers?", he answers: "Yes. That is to say, all the Sculpting gestures – the gestures indicating what content to be performed. Of course, there are gestures such as C Major 7 Chord and Jump that are obviously discipline-specific but most of the gestures traverse the disciplines" (Minors & Thompson, 2012) The multi-disciplinary component of SP is now at the core of its definition, although at first it was designed for musicians. We will see in the part I.A.3 of this report what mechanisms allowed to extend the language to several disciplines.

b) Fertility in Europe, worldwide spread in Western societies, construction of a community and beginning of the normalization of SP

(1) An important spread and growth in Western societies

Thompson gave his first SP workshop in Europe in the late 90's and found a very fertile ground for the growth of Soundpainting in France, which is now probably its largest community over the world; today, Soundpainting is used on every populated continent, especially in the Western world.[2] A community is born around Walter Thompson as well as several SP orchestras on top of the Walter Thompson Orchestra which he started soundpainting with.

(2) A living and expanding language

Several artists created new signs for their own needs with their specific groups and performance configurations: the language has evolved with the contribution of new

[2] I have no example of SP used by artists outside modern societies but experiences with people from cultures that do not share the same linguistic structures and views on the artistic practices and roles would be very interesting for further studies.

soundpainters, now comprising (in 2020) more than 1500 signs. Thompson is still the main figure of SP and gave himself a special role in keeping the language normalized and universal. He leads think tanks, community groups and discussions, the construction of glossaries and dictionaries where the signs, their meanings and uses are discussed.

Two conflictual views of SP are observed: a) On one hand the idea of SP as a language in constant evolution, whose rules and definitions are changing over time as people transform it for their own use; an element of culture that can fundamentally not be owned and controlled. This representation is close to the one of other languages whose diversities and divergent evolutions are well-known throughout history; b) On the other hand, SP is considered as a creation (moreover, with a living creator) that cannot legitimately be transformed by anyone under the same name and whose transformations must be discussed and eventually approved by its creator and the members of the community who participate in the building of its dictionary[3]. Just like various institutions offer certificates for almost every Western language, Thompson proposes certifications in SP "in order to maintain a high level of proficiency for those interested in teaching Soundpainting in an education setting" and maintains an official list of "certified soundpainters" that can be found publicly on his website.[4]

This conflict is also found in several societies between institutions that claim to have an authority on both the syntactic rules of a language and its dictionary and speakers who are constantly transforming the language irrespectively of the approval or control of these institutions.

In his Workbook (thereafter WB) 2, Thompson writes: "In order to address the needs of growth and to keep the language from spreading into hundreds of separate dialects or patois, each year experienced Soundpainters come together to further develop the language in what are known as Soundpainting Think Thanks". The need for normalization and the concern that a language might drift and be so split apart that two speakers would have trouble understanding each other are also found with other sign languages. In France where SP has grown rapidly, a comparison can be made with the divergent views

[3] Thompson has gathered a team of approximately 20 experienced soundpainters to build an official SP dictionary in several languages. Each sign would be described in the system of categories presented by Thompson, with descriptions of what the sign means and how to perform it in text and video.

[4] See http://www.soundpainting.com/certified-soundpainters/. As a side remark, I would like to note that certifications are symptoms of larger paradigms of evaluation, control and normalization in the Western world. These paradigms may be seen at the opposite of any artistic production or teaching, yet they are reproduced and adopted by many artistic institutions and actors themselves.

on the French Sign Language which although being much older than SP, has a number of speakers relatively low who only recently in history started forming a visible community and has very little literature that describes its grammatical rules and structure.[5]

It is often said that "there are no mistakes in SP" and Thompson's view is that "It is much more interesting and challenging to Soundpaint with the so-called mistake than to acknowledge one has been made. [His] experience has been that composing with the mistake is quite often a more interesting direction to take the composition than any [he] could think of." (Minors & Thompson, 2012). But if mistakes are interesting of the side of the performer, is it any different on the side of the soundpainter? We will see that the definition of a "mistake" is conventional, and we will discuss the role of conventions as meta-structures of SP in the section D.3. It is important to point out that languages evolve from mistakes or unintentional deformations.

Soundpainting is presented by Thompson as "the universal multidisciplinary live composing sign language for musicians, actors, dancers, and visual Artists" language (Thompson, s.d.). One may question what is the meaning of the word "universal" in his terms: we will see in B.2 that they are other sign languages in history that have multiple similarities with SP, such that it cannot be considered "universal" in the sense of "unique" and "cross-cultural" but rather in reference to the universe of artistic disciplines, which can all be used with SP. At first, Thompson had protected the name "SP" as his own intellectual property but has then changed his mind and removed its record. This may be a sign of the evolution of his view on these perspectives as well.

c) The revisited orchestras

While SP has developed around a prototypical configuration involving a composer and an orchestra of performers, research[6] in soundpainting and performance has shown the potential of using the language not only for a frontal performance linking a composer and performers but as a communication and synthesis language between the performers themselves and as a interacting language with the public. Thompson himself speaks about the creation of "one handed signs" in SP as the transformation of the language in

[5] For reference, see the article by Le Corre, G. (2007). *La langue des signes française (LSF)*. Enfance, vol. 59(3), 228-236 or the movie « J'avancerai vers toi avec les yeux d'un sourd » (Laetitia Carton, 2015).

[6] For instance, take a look at Conducting with the Body or similar duets, Audrey Vallarino and the Tours soundpainting orchestra, Col·lectiu Free't. Soundpainting (consulted in February 2020).

response to his need for soundpainting while performing, so that he could continue to play while communicating with other performers using SP signs.
In each case, the traditional frontiers between the soundpainter as a composer and the performers as the orchestra are blurry and sometimes irrelevant to describe what I will conceptualize in the rest of this report as different configurations of SP.

B. Theoretical and historical context to SP

To better understand the adoption of SP in the Western culture and the conceptual systems that it is based on, we will start by discussing the theoretical notions of signs, communication and language, before introducing their use in the conduction of music and introducing some other artistic practices similar to SP. We will finish with an aside on cognition, machine learning, the human categorical and prototypical perception scheme that I refer to further in the report.

1. Signs and linguistics

First, I would like to review the notion of sign language in linguistics that will allow us to better understand SP as a language.

a) Theories of signs

In its most basic definition, a sign is "something that stands for something else". The discipline which deal with this notion is called semiotics, and linguistics is basically a sub-discipline of semiotics as our language itself is a system of signs (Brock, Semantics #1 - Signs and Meaning in Language). I would like to introduce a few theories on signs that will help us understand the processes of SP.

(1) De Saussure signified and signifier

Due to his theories on the structure of language, the Swiss linguist, Ferdinand de Saussure (1857-1913) is often known as the founder of modern linguistics (DecodingScience, s.d.). The main idea brought by De Saussure is that a sign is made of two components (the so-called "egg model"):

- A signified, which is the concept or meaning part of the sign (the sign's "content")
- A signifier that represents the signified (the sign's "body")

Moreover, he claims that the relation between the signifier and signified is conventional and arbitrary (it could be something else). This arbitrariness is for instance observed in the diversity of languages in the world: bird as a concept is represented by the word "bird" in english, but also the word "oiseau" in French, etc.

(2) Charles Sanders Peirce

Charles Sanders Peirce, who did not know about Saussure's work proposed at the same epoch a more complex classification of theory of signs in triadic elements (Peirce, 1903). For instance, he claims that signs can be categorized in three classes:

- Symbols: arbitrary signs that must be learned
- Icons: signs for content and bodies that are similar in look, sound, smell or taste
- Indices (or index): signs that are caused by the thing they stand for or bear close connection between body and content

For Peirce, the most versatile signs are symbols because they can express more abstract and complex concepts. It is however important to note the porosity in the frontiers of the classification: a sign is not only a symbol, only an icon or only an indice. These "categories" should rather be understood as three qualities of signs, such that they all are more or less symbolic, more or less iconic, more or less indices.

b) Communication

(1) Elements of communication

Even though more complex models have been proposed since, we find for instance elements of communication in the theory of Shannon (Shannon, 1948), that will help us naming the different processes and operators in SP:

- The source which produces a message or sequence of messages to be communicated to the receiving terminal.
- The sender which operates on the message in some way to produce a signal suitable for transmission over the channel.
- The channel which is the medium used to transmit the signal from transmitter to receiver.
- The receiver who performs the inverse operation of that done by the transmitter, reconstructing the message from the signal.
- The destination which is the person (or thing) for whom the message is intended
- The message which is the concept, the information, or the statement that is sent in a verbal, written, recorded, or visual form to the recipient.

(2) Communication models

Three types of models for communication are generally presented: the linear, the interactive and the transactional.

The linear model represents one-way communication, as a transfer of a message from the source to the destination. The sender encodes the message, for instance with sign language, that is decoded by the receiver. There is no feedback in this model.

The interactive model describes a two-way but asynchronous communication, like message exchanges over the internet. It considers the feedback, the context and notions of behavior for both intentional and unintentional communication.

The transactional model views communication as occurring simultaneously, each person being both a sender and receiver at the same time, even though the language of the communication is not always the same (while someone is speaking, others can react with gestures or unintentional communication forms).

c) Some elements of syntax

Syntax is the part of the grammar of a language which describes the rules that allow to combine elements into sentences. I this section, I introduce some aspect of syntax that will be relevant in our description of SP.

(1) Hierarchical syntactic structures

It is common in Western languages[7] to find hierarchical elements and structures of syntax. For instance, most basic elements of syntax are called "words"; in our case, they correspond to single signs. These elements can be assembled to form phrases. Each phrase performs a single function in a clause, which is the most basic unit of meaning in the larger sentence. For instance, let's consider the following sentences in English and SP:

"My brother, drives; but he doesn't drive, very well."
"Percussions Actor 1, long tone, slowly enter."

As an illustration of these different structures, I have separated each word/sign by a space, a few phrases by a comma[8], each clause by a semicolon, and noted the end of the sentence by a final point. We can see that such structures are hierarchical and that in

[7] Because I am not an expert in linguistics, I would refrain myself from assuming that the syntactic and grammatical structures that we are familiar with in Western languages are universal.

[8] If I had to separate each phrase in the English example, there would be a lot of commas, making it impossible to realize what phrase does each comma correspond to, as phrases are encapsulated in other ones in hierarchical structures.

specific cases, they can fully overlap, so that a phrase can be a whole clause, a clause can be a whole sentence, etc.

(2) Functions

Each element of the syntax performs several functions at different levels, for instance at the clause level or at the phrase level. Whereas a syntactic structure (words, phrase, clause, sentence) is fixed, its function can vary. The most common and basic functions in English and most Western languages are typically "verb", "noun phrase", "noun", "object predicative", etc, but the set of possible functions depends on the considered language.

d) Language, langue and parole

There are several understandings and definitions of a language. According to the Collins dictionary[9], the word "language" can mean:

- "A system of communication which consists of a set of sounds and written symbols which are used by the people of a particular country or region for talking or writing". In that case, the context of a country or a region is pointed out.
- "The use of a system of communication which consists of a set of sounds or written symbols." In this second meaning, language appears as a system out of contextual elements. It is often used to mean the human faculty of communication and use of symbols in general.

De Saussure however formulated with french the words *langage*, *langue* and *parole* three distinctions: *langage* for language as a concept, *langue* as a specific instance of a language system and *parole* for the concrete use of speech in a language.[10] Only the notion of parole in his terms embodies the idea of a context; *langage* and *langue*, both translated in english as language, refer to a formal system of signs governed by grammatical rules.

In the following of my report, I will push for a structuralist and linguistic approach to SP, in which grammar plays an important role and the idea of a language is decoupled from the contexts in which it is used, such as the area, people who practice it or its uses. In my terms, the word language will therefore follow Saussure's meaning of *langue* and *langage.*

[9] See https://www.collinsdictionary.com/dictionary/english/language

[10] Source : https://en.wikipedia.org/wiki/Language#Definitions

e) Context-free, context-sensitive and regular languages

In order to introduce the parsing mechanisms of my recognition tool, I would like to explain some of the basic concepts of context free/sensitive grammars and explain their link with automata.

(1) Parsing

Parsing, also called syntactic analysis, is the process of analyzing a string of symbols (or sentences) that are described by a grammar. It can be achieved by abstract machines called automata, of different types, depending on the grammar that is considered. One of the results of parsing is the identification of the types of syntactic structures and their roles in the sentence.

(2) Context sensitive grammars

Context sensitive grammars describe the grammars of oral and written languages we are familiar with, such as English. As its name suggests, it means that to find the function of an element in the sentence, it may be necessary to know about the context, i.e. the other elements that surround it. One important theorem is that all context-sensitive grammars are described by a linear bounded automaton (LBA) that could for instance be implemented on a Turing machine such as a computer (Hopcroft & Ullman, 1979).

(3) Context-free grammars

At the opposite of context-sensitive grammars, context-free grammars describe languages in which it is not necessary to know about the context to find the function of an element. Context-free grammars are therefore a subset of context-sensitive grammars. For instance, consider the grammar made of the following production rules:

S -> Who What How When
Who -> identifier
What -> content
How -> modifier | **ε**
When-> go gesture

Where "Who", "What", "How", "When" are non-terminal symbols (representing syntactic structures in the grammar), "identifier", "content", "modifier", "go gesture" are terminal symbols representing words or signs in the sentence, S is the start symbol (sentence symbol), **ε** the empty string symbol and | the "or" logical operator. This grammar would only produce two different sentences:

- "identifier content modifier go gesture"
- "identifier content go gesture"

Because the production rules on the left side only contain one element, it is possible to find the function of each element in the sentence without the context. This grammar is therefore a context-free grammar.[11] One important theorem is that all context-free grammars can be described by a pushdown automaton, which is a finite state automaton with an infinite “stack” attached. Moreover, all context-free grammars satisfy the necessary (but not sufficient) pumping lemma for context-free grammar. Simplified, this lemma states that sufficiently long sentences always have a “recursive” part, i.e. a part that is repeated inside another.[12]

f) Linguistics in generative music

The contemporary field of generative music found interest for many models from linguistics to describe and generate music. Often implemented in computers and used for experimental music, real-time jazz accompaniment and improvisation to Beatles-like song production, probabilistic models such as Hidden Markov Models are increasingly being used these last decades in research and recent production and composition software, such as Max/MSP.

g) Max/MSP

Max/MSP, or simply Max, is a visual programming language for music and multimedia created at IRCAM in 1985 and today commercially maintained and developed by the company Cycling ’74. Max was first used to send control messages to external hardware synthesizers and samplers using MIDI or a similar protocol but was later extended with Max Signal Processing (MSP) to handle real-time digital audio signals without dedicated DPS hardware. Later, Cycling '74 released their own set of video extensions, *Jitter*, alongside Max 4 in 2003, adding real-time video, OpenGL graphics, and matrix processing capabilities.[13] Max is a popular tool for interactive and media art since its early days.[14]

[11] This example is only a very simplified example of the SP grammar. One should not conclude at this point that the SP grammar is context-free, nor regular.

[12] For details about the exact lemma, see Wikipedia: https://en.wikipedia.org/wiki/Pumping_lemma_for_context-free_languages

[13] Source : https://en.wikipedia.org/wiki/Max_(software)

[14] One of the pioneers of interactive art, David Rokeby, released a set of video extensions that he used in one of his early work *Very Nervous System* (1982–1991), presented at the Venice Biennale in 1986.

2. Gestural signs for communication: a long history

In this part, we will have a short overview of the use of gestural signs for communication in history and highlight the different motivations for using gestures in music and artistic contexts.

a) Cheironomy

Cheironomy is defined by Hickmann in 1949 as “textually the direction of a musical ensemble by the movements of the hand” (Huglo, 1963). It is employed to refer to the use of controlled, regular and organized gestures that is mostly encountered in texts about the arts of movement: music, dance and pantomime. For instance, in the modern artform, conductors tend to hoist batons for indicating melodic curves and ornaments.

(1) First traces in antiquity

We know from sculptures and paintings from ancient Egypt (at least 4 000 years ago) that they used a form of cheironomy to indicate several pitches and rhythms, sometimes as they were singing, although singer and cheironome were in principle two different roles.

In Greece, ascendant and descendant movements of the hand were also used during antiquity to indicate whether the next note was higher or lower in pitch. Cheironomy was then considered a form of art (Huglo, 1963).

(2) Cheironomy in middle ages

Cheironomy had an important place for the direction of Gregorian chant from the middle ages to the 16th century. Gregorian chant has extensively used the notation system of neumes. In fact, the most ancient neumatic notations of music were also called

"A combination of technologies, some off-the-shelf, some rare and esoteric, and some cooked up by Rokeby himself. Initially, in 1982, much more of the system was homemade. His circuitry, designed to speed up the response of the sluggish Apple II, was still not fast enough to analyze an image from an ordinary video camera, so he built his own low-res device: a little box with 64 light sensors behind a plastic Fresnel lens. But Very Nervous System has been evolving for 13 years, during which time the world has seen any number of technological revolutions. So Rokeby now has a lot more store-bought components incorporated into the system: it can handle a Mac Quadra and real video cameras, via sophisticated "Max" software from Paris."
Source: Douglas Cooper, "Very Nervous System: Artist David Rokeby adds new meaning to the term interactive", *Wired* Issue 3.03 (1995)

cheironomic notations and the grec “neuma” refers to the latin “nutus” according to the grammarian Comminianus and was used prior to the notations of music on parchment to signify a vocal exercise. Michel Huglo makes the conjecture that cheironomy was in fact the ancestor of the neumatic notation (Huglo, 1963).

In the 11th century, the famous “hand” of Guido d’Arezzo is another example of musical cheironomy during the middle ages. These systems were used as learning methods and as memorization helpers. Later, they will be abandoned for written notations, but they are still part of the musical learning in some regions of the world such as India and may be considered the ancestors of today’s conduction techniques.

b) Sign languages in the deaf communities

Sign languages are most known for their extensive use in deaf communities. Their emergence among deaf is reported several times from the XVIIIth century to the XXth century.[15] Old French Sign Language is one of the first sign languages that was reported in the deaf communities and is a partial ancestor to many other sign languages including the American Sign Language. It has however been forbidden since 1880 and only officially re-approved in schools in 1991: sign language was thought as a primitive form of communication and was not known from the public in France until the late XXth century.

(1) As an artistic element

As an artistic element, the sign languages - that come from deaf communities - are also used outside of this community as a way of exploring new expression means with the body; indeed, they make extensive use of facial expressions and iconic gestures, similarly to theater, dance and other artistic disciplines.

(a) In theater and dance

Inside the deaf communities themselves, several artists use their sign language as an expressive and creative element. In 1977, the deaf American artist Alfredo Corrado created the International Visual Theater in Paris to lead research in non-verbal theater and sign language as artistic tools.[16]

[15] For a striking example of the emergence of sign languages, see *Children Creating Core Properties of Language: Evidence from an Emerging Sign Language in Nicaragua* By Ann Senghas, Sotaro Kita, Asli Özyürek, Science (2004).

[16] See the IVT’s website http://ivt.fr/ for further documentation about the history of its creation and resources on artistic practices with sign language.

(b) "Sign singing"

Recently, several artists such as the americans Sean Forbes and Brandon Kazen-Maddox[17] started creating and adapting songs in sign language, called "sign singing" in English or "chansigne" in French. Lyrics may be translated in sign language but already composed in sign language directly. It is now taught in a few places, such as the "Opéra comique" in Paris as a discipline in its own right.

c) Monastic sign languages

Monastic sign languages have been used in Europe from at least the 10th century by Christian monks and are still in use today, not only in Europe but also in Japan, China and the USA. Unlike deaf sign languages, they are better understood as simple signs, lexicons and forms of communication that could be used when silence was required or as memory aids rather than languages (Barakat, 1975 and Quay, 2001).

d) Earle Brown's Open Form

Earle Brow composed a series of "Open Forms", made of fixed modules, structured in "events" and divided in "pages" whose order was chosen by the conductor during performance. "The conductor uses a placard to indicate the page, and with his left hand indicates which event is to be performed while his right hand cues a downbeat to begin. The speed and intensity of the downbeat suggests the tempo and dynamics".[18] As Thompson (although years before), Brow spent years at what is now the Berklee College of Music and was very influential in the New York scene, where he collaborated with John Cage among others.

e) Conduction

Conduction is often reported by Thompson himself as one of the closest systems to SP. In his work "Soundpainting as a system for the collaborative creation of music in performance" (Duby, 2006), Duby introduces Butch Morris' Conduction as a system of gestural signals:

[17] Sean Forbes is a rapper famous in the American deaf community. He interprets his songs in sign language: Sean Forbes - I'm Deaf (consulted 13/06/2020). Brandon Kazen-Maddox is a dancer, choreographer and American sign language interpreter who leads a production company specializing in producing, directing, editing and collaborating alongside the deaf community (https://www.youtube.com/channel/UCSS8DcHaFJeIWOGe2nGB8bQ).

[18] Source and quote from https://en.wikipedia.org/wiki/Earle_Brown#Open_form

> The New York cornetist, Butch Morris, has also developed a system of signals for musical purposes. Morris's system, known as conduction, has been exhaustively documented (notably in Mandel 1999), and depends on a much smaller number of gestures (around 30) than Thompson's (around [1500 in 2020]). In an interview (Mandel 1999:65), Morris describes conduction, not so much as a language in Thompson's terms, but as a 'gestural vocabulary'.

On the contrary of SP, Conduction does not allow for forming structured requests using a particular syntax but rather to extend the meaning of the gestures and movements of the baguette with symbols and icons that are more meaningful than usual icons in traditional conduction. For a comparative discussion of SP and Conduction, I recommend "Problemas de performance em improvisação dirigida: um estudo comparativo dos sistemas de Soundpainting e Conduction®" by Peluci de Castro (Peluci de Castro, 2015).

f) Ritmo y Percusión con Señas

More recently, we have seen the emerging of other signs systems; I would like to mention "Ritmo y Percusión con Señas" that derived from Conduction and specialises in percussion groups (La Percumotora):

> Rhythm and Percussion with Signs is an innovative way to play percussion created by the Argentinian musician Santiago Vázquez, who was Inspired by Conduction [by Morris]. Santiago was looking for a way to communicate certain information to musicians improvising percussion, in order to generate coordination and harmonization of a spontaneous creativity. Through these attempts, little by little he codified a language consisting in approximately 150 signs executed with hands and body by the conductor in order to coordinate the flux of group improvisation.

3. Artistic context of SP and computer recognition tool

SP and Thompson's practice enters in the wider movements of music and arts of their generations. In this section, I would like to superficially explore the artistic landscape in which SP lies (focusing on music only) and identify in computer music the potential of linking SP to machines.

a) Artistic context to SP

In music, the 20th century can be thought of as the century of emancipations, divergences and exploration of new forms of performances and productions.

We can identify at the time of the development of SP a great number of esthetics in the production of music but also a specific interest for new production processes, such as aleatory (John Cage), serialism (Arnold Schoenberg) and stochastic (Iannis Xenakis) music. Such compositional techniques were sometimes implemented with machines such as computers (electronic, concrete music, etc), for instance in France by Pierre Schaeffer and Pierre Boulez.

In jazz, the rupture with the "traditional" rules created the new "free" jazz whose influential figures are for instance Anthony Braxon who Thompson studied with and François Jeanneau, a pioneer of French free jazz who is now a leading figure in SP.

b) Computer music

Computer music is the use of computing technology in music composition, from sound synthesis to algorithmic composition programs. Today, the widespread availability of relatively affordable Digital Audio Workstations (DAWs) and the growth of home recording studios has brought computer music everywhere in the landscape of music production and diffusion. Aside from the notion of computer music for composition only, the use of digital tools for recording, processing and diffusing sound make computers the most widely used interface for music creation. The potential of computers and digital tools is to my opinion still not expressed in SP and the construction of a recognition tool could be an important step in that way.

4. Cognition and classification in Machine Learning

The word "cognition" dates back to the 15th century, where it meant ''thinking and awareness.'' (Revlin). The term comes from the Latin noun *cognitio* ('examination,' 'learning,' or 'knowledge'), derived from the verb *cognosco*, a compound of con ('with') and gnōscō ('know'). The latter half, gnōscō, itself is a cognate of a Greek verb, gi(g)n**ó**sko (**γι**(**γ**)**νώσκω**, 'I know,' or 'perceive') (Liddell, Henry, & R., 1940). Re-cognition can therefore be understood as the process of "another" cognition, that allows the identification of something already known or already perceived.

One can wonder what the core processes of recognition are and try to reproduce them on a computer. In machine learning and statistics, classification is the problem of identifying to which of a set of categories a new observation belongs, based on a training set of data containing observations whose category membership is known. It can be considered as a recognition mechanism, and it is typically achieved with a supervised learning procedure. A supervised learning procedure comprises 3 phases:

- First, a ground truth dataset is constructed from our prior knowledge of what output values for our samples should be. The dataset comprises one or several training examples.
- Then, an algorithm is used to “learn” iteratively a function that, given a sample of data and desired outputs, best approximates the relationship between input and output observable in the data. Several algorithms can be used for this process.
- Finally, the “learned” function that we call “model” can be used to classify new data.

Some classification models require training on huge amounts of data, for instance deep learning models. These models typically perform as well or better than humans for specific tasks. Some light-weight models however might require only a few training examples but are often less performant.

5. The human prototypical and categorical schemes of perception and concepts

The human categorical perception scheme plays a very important role in the construction of basic artistic concepts such as note, pitch, scale, line, hit… by constructing discrete categories out of a continuous set of elements (frequency, time, space, color, etc). One can think of these schemes, that contain innate and learned (cultural) parts, as processes of identification and classification.

We know from research in psychology (Rosch, 1973) that a single concept can be modeled as a category of elements around a prototype, considered as the central point of the category. Moreover, people tend to define the concept itself by the characteristic traits of the prototype, whereas in general, it extends beyond such a definition: the prototypical scheme rejects the discrete notion of ‘limit’ or ‘border’, replaced by the continuous notions of graded membership (similarity to the prototype) and the fuzzy edges of concepts (Brock, Semantics #4 - Prototype Theory). We have seen earlier that a sign does not fall exactly in the category “symbol”, “icon” or “indice” but is rather “more or less” of each of these qualities. It is the same with concepts in general: humans tend to perceive things as “more or less” belonging to a category or a concept. For instance, US people consider ostriches as “less a bird” than the American robin which Rosch found out to be the most prototypical (i.e. the most representative) example of birds in her studies.

But on top of the prototypical, continuous scheme, the categorical scheme introduces a rupture by either accepting or rejecting an element inside the category based on its similarity with the prototype.

a) Remarks on analogy

In his conference “Analogy as the core of cognition” (Hofstadter, 2009), Hofstadter defines making an analogy as raising the similar features of two mental things. As we talk about the categorical scheme of perception of humans, it is important to mention analogy as responsible for the extension of a concept from its prototype and as the main process behind the evaluation of similarity:

> Categorization is the name of the cognition game and analogy is the mechanism that drives it all.

b) A side-note on the conceptual space

In all previous discussion, I have considered the idea of boundaries between concepts, just like if we could model the space in which they lie as a 2D or 3D space where they appear as clusters of points around a center. In fact, this model is only valid as long as we consider each one individually. When describing the combination of concepts, we can no longer think of them as such. For instance, the German way of building words is a very explicit example of the hierarchical construction of some concepts. Whereas it is also clear that they are not only hierarchical structures either, they do embed some form of hierarchy, such as studied in research in ontologies in the field of Digital Humanities (Dau, Mugnier, & Stumme, 2005).

As a final word, I would like to mention the quantum-like formalism as one of the most promising description of the way concepts interfere with each other, although they have no explanatory power[19] (Aerts, Broekaert, Gabora, & Sozzo, 2016).

C. A structuralist approach to Soundpainting

Now that we have seen a bit of the development of SP and the theoretical background it lies in, I would like to propose a model of Soundpainting that explicits its construction mechanisms as a sign language, as well as the implicit operations that makes it an efficient language for art performance. My approach is a structuralist one: I will focus on the structures of SP at several levels, in a bottom-top approach rather than the description of the constitutive elements themselves.

[19] Indeed, the mapping from concepts to vectors in the Hilbert space is not derived from first principles, but rather inferred a posteriori by fitting the predictions of the model to the experimental data, thus removing any explanatory power.

1. Preliminary remarks: context and scope of my personal observations

The reader must be aware that this section is mostly based on my personal observations of and participation in SP performances, mostly as a performer with only a limited number of groups (7) during the past 7 years (most of my experience however comes from the last 3 years). Moreover, I have a unique experience of SP outside Europe (Brasil), while my experiences in Europe are in France and Switzerland. While I will try in the following sections to speak about SP as objectively as possible, I will use the first-person pronoun to indicate my own analyses and points of view. The reader is invited to compare my observation with his and criticize the models and interpretation I give.

2. Mapping of concepts from different spaces onto signs in the physical space of/around the body: the origin of SP signs

When we speak of sign language, we usually speak of gestural sign languages, in which the signs are expressed with gestures. In linguistic terms, all languages are sign languages. In this section, I would like to discuss the formation of signs using gestures in SP as a mapping of concepts from different conceptual spaces onto the physical space of the body (Figure 1).

In mathematics, a mapping[20] is often described by a function from a domain (the input space) to a codomain (the output space). In the case of SP, I would like to identify several conceptual spaces from which it constructed its own concepts (several input spaces) and identify several output spaces that may be signs with the body but also imaginary spaces around it that are significant.

[20] Here, I use the term mapping as a general term, without figuring out whether it can more precisely be called a transformation or projection in mathematical terms. However, in the common sense, one can understand it as a transformation or a projection.

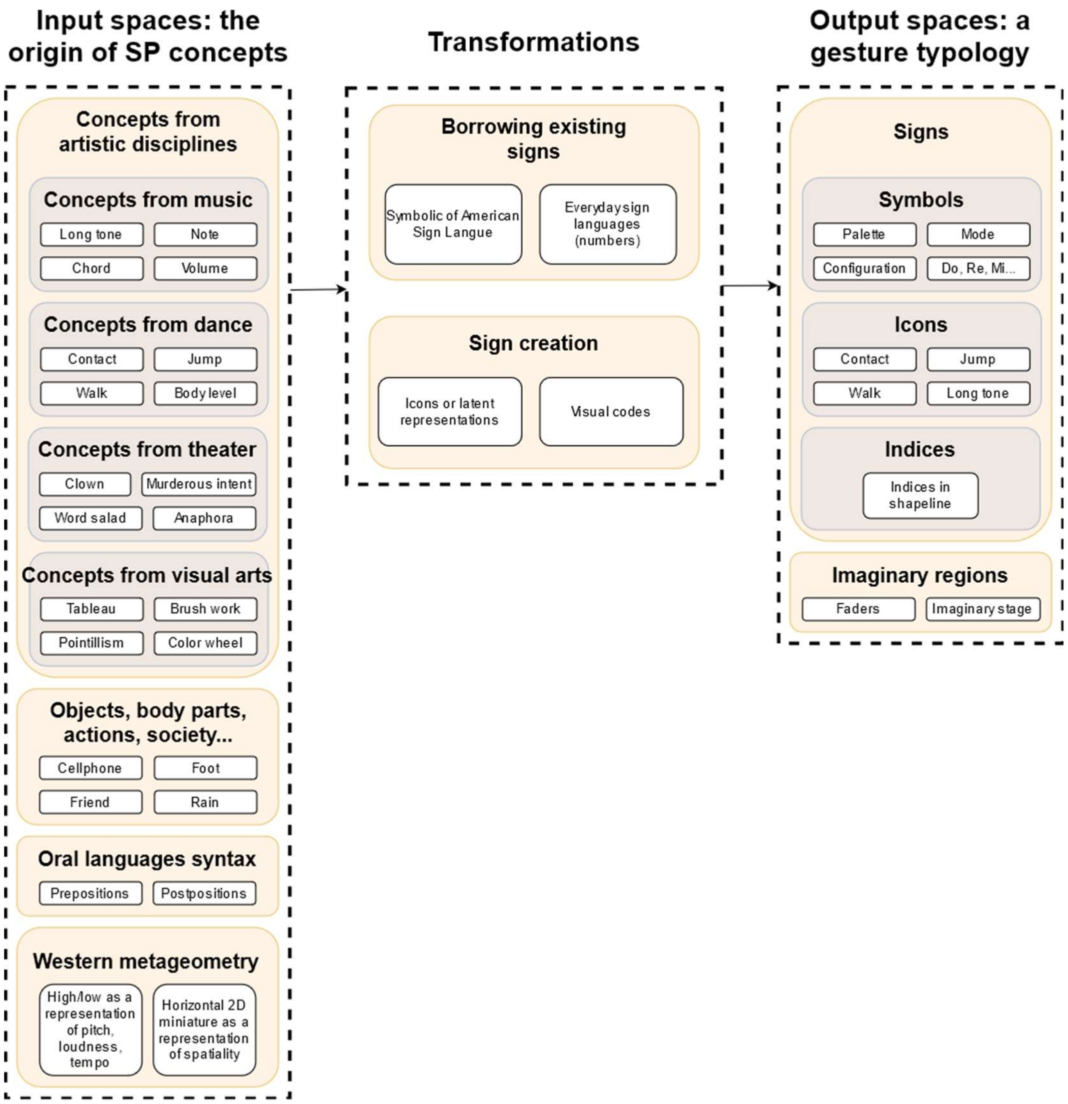


Figure 1 SP as a transformation from several conceptual spaces to different gesture types and imaginary regions

a) Input spaces

We can identify several repertoires (sources) of concepts as different input spaces of the mapping scheme of SP.

(1) Concepts from artistic disciplines

The first input space that I would like to consider is the space of concepts from existing artistic disciplines. As Thompson says (Minors & Thompson, 2012), “I created many of the gestures from concepts found in theatre, music, dancing, visual arts […].” In those cases, we can usually identify the origins of the concepts from their names: “Long Tone” (from music, although “tone” itself could also refer to visual arts), “Brush Work” (visual arts – painting), “Tempo” (music), “Volume” (music), “Snapshot” (visual arts – photography), “Hit” (dance, music), etc.

(2) Cultural representations of pitch

However, SP also has several gestures that are born from cultural representations proper to Western context, such as the concept of height for describing a sound (e.g. high and low). Even though the concepts of volume, tempo or pitch may be universal, their mapping onto a low/high axis is part of the metageometry, (i.e. the rationalization of the spatial condition of knowledge) of modern societies, that use space and time (we already saw the example of cheironomy in B.2.a)) to represent relations in music. In fact, the music intelligibility of qualities such as the frequency of sound through representations in space is common to many musical civilizations but the choice of particular orientations in space is arbitrary and varies with cultures and history (Duchez, 1979):

> Ni arbitraire, ni naturellement perçue, la hauteur du son est – l'histoire du concept le montre - le résultat d'une construction rationnelle, à partir d'une perception primitive préférentielle sans spatialité, la qualité grave-aigu, sur laquelle elle est superposée, et que, grâce à l'éducation, elle transforme en perception conceptuelle. Cette construction rationnelle, bien que très répandue, n'est ni générale, ni, sans doute, définitive.

Duchez argues that already in ancient Greece, one would speak about voice with movements, about the location of a sound or intervals as a distance between sounds: there was already a clear notion of spatiality in the descriptions of sounds. However, it is in the 9th century that the rationalization of pitch in terms of height appeared with the prominence of Gregorian chant and later became systematic with the development of the notation systems (neumes) and in the pedagogy after the 10th century.

Similarly, the concept of height for representing tempo, volume and sound in general are historical constructions that do find universal basis (sound as a spatiality) but whose precise orientations and projections on the vertical axis are culturally defined conventions.

(3) Objects, body parts, actions, society…

To the question "In what way would you say your gestures are culturally determined (specific to Western contexts)?", Thompson responds (Minors & Thompson, 2012):

> The gestures are culturally determined in a sense that their physicality is created from what we see in everyday life such as events occurring on television, computers, people in crowds, sports, etc.

While we showed in the previous point that they were not the only culturally determined elements, SP does indeed borrows concepts from the Western culture that refer to objects, body parts or actions: "Cellphone", "Drone", "Tear up", "Chair", "Cops", "Change This", "Blinders", "Prepare", "Sprinkle", "Mouth", "Open", "Close", "Woman", etc.

(4) Elements of syntax from the grammars of Western oral and written languages

Additionally, SP borrows some elements of grammar from Western languages, in particular adpositional elements (prepositions and postpositions): “with”, “within”, “without”, “through”, “go back to”, “go onto”, “enter next cycle”, etc. We will see in a future section that the instruction of these elements fundamentally changes the SP from a context-free to a context-sensitive grammar.

b) Output spaces: gesture typology

Now that we have described the main input spaces from which SP borrows or construct concepts, I would like to identify a typology of gestures:

- three types of signs (according to Peirce’s classifications of signs):
  - symbols
  - icons
  - indices
- the imaginary faders that do not exactly act as a sign but rather as a underlying scheme for signs which represent space or frequency
- the imaginary box that acts like punctuation marks and a meta-sign for giving meaning to other signs

It is implicit that all SP gestures do not enter exactly in one category only; there is at least a symbolic component to each sign, while they may also carry several elements of iconicity (see B.1.a).

Moreover, in the case of symbols, I will distinguish the transformation processes of creation of a symbol and borrowing of a symbol from another language.

(1) Symbols

(a) Creation of a symbol

Symbols in SP may appear as the less convenient type of sign (despite Peirce’s argument of versatility): because they do not convey an image of what they represent and are mere conventions, they may be difficult to memorize and transform. Ideally, SP signs are therefore constructed with as little “symbolicity” as possible, i.e. that the scope of their arbitrariness and conventionality is kept as small as possible.

There are still a few symbols created in SP, for instance “Palette”, “Mode”, “Configuration”, the signs for each notes and chords, “Melody”, etc, in which I could hardly see any icon or indice.

(b) Borrowing of a symbol from another sign language

It is interesting to note that very few symbols in SP come from other sign languages such as the American Sign Language (ASL). “Bullshit”, “Man” and “Women” are some examples of signs borrowed from ASL.

(2) Creation of an icon

In his interview with Helen Minors (Minors & Thompson, 2012), Thompson says:

> Some of the gestures I created are quite iconic such as Long Tone and Pointillism and others have little or no visual relation to the material to be performed. I created the Play gesture from the movement used when Bowling – the release of the ball. I modified the movement to incorporate two hands. Many other gestures share a similar history. I may also add [that] the gestures of the Soundpainting language are not [usually] based on the language for the deaf or hard of hearing.

Duby also remarks that (Duby, 2006):

> The iconic element of the gestures of Soundpainting is often conveyed in a humorous manner. For example, the “Rock” gesture, in which the soundpainter clenches the fist (to mimic throwing a rock), can be interpreted quite literally "to mean what it says," but at the same time the punning visual element in this gesture lends it an ironic, tongue-in-cheek quality. In this sense, many of the gestures have a double meaning, functioning both as icons and as ironic reflections of an unproblematic iconicity.

As additional examples, one can think of “Break”, “Change” and “Volume” (as the icons for the letter “C” and “V” of “change” and “volume”), “Match”, etc, as very iconic signs.

One should remark that icons are very performative: slight changes in the icon are often meaningful and provided subtilities to the sign; both their learning and memorization are made easier by the clear reference they provide to the concept they signify. Whether it is intentional or not from their creators,, icons are indeed the most important and widely used type of signs in SP. We can at this point remind ourselves that icons are efficient without prior convention, therefore the emergence of SP in emergency could only allow Thompson to use very iconic signs if he expected a response from the performers.

(3) Use of indices

Indices are rather difficult to observe in SP (I think they are the less usual type of sign among those that are commonly use to describe SP) and I can here only point to their use in “Shapeline”, a SP grammatical component[21] in which all signs – not only SP signs – must be interpreted by the performers, allowing the soundpainter to use indices (just like an actor),such as facial, body expressions or marks. To my knowledge, there are no indices in SP signs outside those used in shapeline.

(4) The imaginary regions

(a) The imaginary staff

In his website (Thompson, s.d.), Thomson introduces the “imaginary staff”:

> The Imaginary Staff: An imaginary vertical field 1 and ½ meters (3 ½ feet) just in front of the Soundpainter that indicates low to high pitch range with sound and slow to fast movement with certain gestures such as a Long Tone. Note: The name Imaginary Staff is derived from music language. It is related to the music staff, which is a set of five parallel lines with spaces between them, on which notes are written to indicate their pitch.

This region is an explicit reference to the representation of pitch in terms of height in the Western culture, which Duchez refers to as an element of the metageometry of Western culture.

(b) The imaginary stage

Thompson also describes an “imaginary stage” as the following (Thompson, s.d.):

> The Imaginary Stage: A horizontal field (like a small square table top) approximately ¾ of a meter for each side (3/4 of a yard squared) at waist height positioned just in front of the Soundpainter. The Imaginary Stage is the region in which the Soundpainter indicates movement directions on the stage – where the movement will travel to and from. Such gestures as Directions and Space Fader are both signed on the Imaginary Stage.

This time, it is not a representation in terms of height that is implied but rather a 2D miniaturization of space and the cultural concept of stages that are invoked in his description.

(c) Imaginary staff and stage as faders

These representations - imaginary staff and stage as faders - share common traits:

- They are the result of at least three operations:

[21] Such grammatical components of SP are commonly called “modes”.

    - spatialization (or re-spatialization in the case of the stage), that one could also call a projection
    - orientation
    - evaluation
- Their relative character (there is no absolute value or measure in each representation), therefore they function as a relational space
- Their function as underlying scheme for several signs

Similarly, to the frequency of the sound, tempo or volume, height as a spatial evaluation is the basis of many "faders" that can represent complexity, exaggeration, age, amplitude, body level, gravity, etc: faders are the main type of imaginary regions in SP.

In the case of faders related to notions of space, such as the density fader or the "more space" fader, it is interesting to note the choice of the horizontal rather than the vertical axis, following the horizontal metageometry of space in Western cultures.

(5) The neutral position and imaginary box as punctuation marks and meta-signs for meaning

Finally, Thompson describes a dichotomy between two imaginary spaces that are called the "neutral position" and "the imaginary box", which I will refer to in the future simply as the box. These regions are not faders but specific signs, that I consider the equivalent of the punctuation in written languages: the neutral position is where the clauses and sentences are formed, whereas the box indicates the end of the sentences and the clause (see B.1.c), i.e. communicates that the sentence is meaningful when the soundpainter steps in the box. One very important part of the construction of the box is that it can be used simultaneously to other signs, so that the sentence can immediately be understood as finished, as one would expect in real-time communication. It is the equivalent of punctuation in written languages or intonation in oral ones.

The box can therefore be considered as a kind of "meta-sign" that gives meaning to other signs.

c) A lead for understanding the mapping process: visual codes

From the typology that I have introduced, one may wonder at a more fundamental level what the mapping process relies on and how exactly the sign is constructed visually by its creator and understood by other people. As a first answer to this complex question, I would like to point out the analysis of SP visual codes by Yerlikaya and Coskuner (Yerlikaya & Coskumer, 2016), that brings finer element of interpretation to the signs that explicate how they may have been formed intentionally or not by their creator. For instance, they analyze the “knowledge of visual contract” - contract in the sense of internalized visual rules and meanings - in the sign “Whole group” as “Circle [that] evokes the concepts like a round table or a ring. It incorporates the constructs as whole, being together etc.” among several visual codes that are relevant to explain the signifier-signified relation.

3. Sign overloading: polysemy and technical approaches to concepts in SP

In Minors’ interview of Thompson (Minors & Thompson, 2012), she asks:

> Pitch is reliant on a Western concept of height in relation to sound (e.g. high and low), replying on our metaphorical understanding of music. How would you expect a dancer/actor to respond to this musically determined gesture?

Thompson responds with the following:

> Pitch is a frequency and its tempo governs whether the sound is low, middle, or high. Movement takes place in space and is also governed by tempo. When a musician is signed a Long Tone (middle range) they choose a pitch from this region, a dancer signed the same Long Tone will perform sustained movement at a medium tempo. The Long Tone gesture spans all the disciplines. I created the physicality of the Long Tone gesture and just like any spoken language each performer must learn what the gesture means. Being a musician, dancer, actor, visual artist, etc. does not present any discipline-specific problems when learning the signs.

Despite his questionable definition of pitch and movement as governed by tempo, we can read in his words the analogy underlying the significance of the notion of pitch in both music and dance. We have seen that SP is a multi-disciplinary sign language, i.e. that a single sign can be used to signify a content[22] not only for (1) different instruments of the same discipline but also (2) across discipline. This is what I call the “overloading” of a sign[23]. In this section, I will try to show that:

[22] Note that the whole discussion of this section is only relevant to the signs used to signify contents.

[23] In reference to the concept of overloading in programming languages.

- there is a polysemy of signs across disciplines: each sign refers to a different concept in each discipline (at the contrary of Thompson's statements)[24]
- the polysemy of multi-disciplinary signs is constructed from analogies between different concepts in each discipline
- disciplines are conceptually different from instruments
- the operability of a sign inside a discipline – with different instruments – is a technical operation of translation of the concept of their discipline to the set of producible material realized by the performers

Therefore, the overloading of signs in (1) and (2) involves mechanisms of different nature.

a) Polysemy of signs across disciplines

We commonly observe signs made of one signifier and several signified in oral languages and in our everyday life. When one signifier can stand for several signified, the meaning of the sign is found by an operation of "disambiguation" that depends on its context. We call the fact that a sign can have several meanings "polysemy" and such words "polysemes".

In this part, we will see that multi-disciplinary signs are the main polysemes in SP, having a different meaning to each discipline[25]. Then, we will discuss the construction process of these multi-disciplinary signs to show how their signifieds are linked by analogies.

(1) Multi-disciplinary signs

To demonstrate the existence of several concepts[26] under a multi-disciplinary sign in SP, let's take the common example of the long tone, thereafter "LT", that we will use all over this section to demonstrate some of the mechanisms of SP.

[24] "When first teaching Soundpainting to a group of musicians and dancers I explain that almost all the gestures, except for a few discipline-specific ones, mean exactly the same concept or something similar (equivalent) for each of the disciplines." (Minors & Thompson, s.d.). Here, I would agree that they mean something similar (by analogy) but not exactly the same concept.

[25] There may be other cases in which a sign has several signified, even inside one discipline, but we won't discuss this possibility.

[26] Here, I use "concept" as an equivalent of "signified" or "meaning".

For a musician, the LT is a concept preexisting to SP with a specific prototype, whose main characteristic traits are “constant volume”, “constant pitch”. But is the concept of the LT for a musician the same as the one for a dancer or a visual artist? To answer this question, let’s first remark that soundpainters often explain how to perform a LT differently for each discipline, often illustrated by a prototypical example:

- “A fluid movement, without accent” for dancers
- “A freeze on the first syllable of a word” for actors
- “A note with constant volume and constant pitch over time” for most musicians...

By looking at the description themselves, we can see that they involve different concepts: a “movement”, a “roll” or a “syllable”, which cannot be considered equal. Moreover, we know from the history of SP that the concept of a LT was first borrowed from music and “extended” to other disciplines, i.e. that the multiplicity of signified of the sign “LT” is a voluntary construction[27]. Although we illustrated the overloading mechanism with the sign “LT”, we can observe the same mechanism for all multi-disciplinary signs.

b) Analogies as the core of polysemy

We have seen previously that the concepts we are dealing with when referring to a LT or other artistic concepts are the result of a mental classification process. Moreover, we also know from Hofstadter (Hofstadter, 2009) that the similarities between concepts are expressed with analogies.

In Thompson’s description of the LT (see I.A.3 above), we can read his interpretation of “tempo” as a common feature between pitch and movement[28]. In other descriptions of the LT, one would also find the “esthetic constancy in time” as another possible common feature between them, among others. Moreover, these features are in general cultural, learned ones rather than innate features from our human perception scheme. In his words, it is because of these common features that a movement in dance and a pitch in music can be associated: he forms here an analogy.

[27] We will discuss the motivations of this construction later.

[28] By « tempo » he his implicitly referring to the frequency of a pitch and the speed of a movement; one could easily argue that frequency and speed can only be compared in the context of a cyclic movement, still the power of analogy relies in its ability to associate elements that are not directly linked.

The same principles are found with other multi-disciplinary signs, such as “volume” (analogy between the loudness in music, extension of the body in dance, sizes or dimensions in visual arts), “minimalism” (analogy between the repetition of forms, colors and traits in visual arts, of notes and rhythms in music, of movements in dance), “complexity”, etc.

(a) Long tone viewed from psychology

Now that we have identified analogies as the core of the polysemy of signs across disciplines, I would like to shortly comment the way a concept (in SP or elsewhere) is usually presented in the form of a definition from the point of view of psychology. It would be tempting to define the long tone for each discipline by giving a set of characteristics that all long tone must have, for instance:

- In music, a constant pitch and volume over time
- In dance, a movement without accents…

About the definitions themselves, we know from psychology (Rosch, 1973) that they are not representative of the entire concept (and usually, one can find counterexamples to the definition). A LT is not and cannot be defined with such characteristics or traits: one has to learn what is a LT through practice and exploration of the concept. The role of the soundpainter here is to introduce a prototype of the concept (the LT) for each discipline by giving one or several examples of what it can sound or look like, while the concept will expand in the performer’s mind by constructing materials analogical to the prototype.

One interesting conclusion is that there is no “wrong” long tone; in this framework, a content is perceived as “more” or “less” a LT, rather than either a LT or not. Essentially, we can also explain the fact that there are different concepts and prototypes in each discipline by the difference in phenomenon: a sound, a movement, a visual, etc. as each independent concept that cannot be explained by the others.

c) The operability of signs inside a discipline, across instruments

In this part, I would like to discuss the operability of signs across instruments of the same discipline. Although it is the polysemy of multi-disciplinary signs that allow it to be significant in several signs, I will argue that across instruments of the same discipline, a unique concept, represented by its prototype[29], is approached by different technical interpretations, possibly involving the human perception scheme. The term “instrument” is used here as an equivalent of “technical apparatus”.

(1) Preliminary remarks about the instrument as technical apparatus

I would now like to remark that a technical apparatus can be characterized by the set/range of artistic material it can provide. It is not obvious that every instrument (as technical apparatus) can achieve similar concepts: monophonic instruments are not able to perform a chord, resonating instruments can hardly perform staccatos…

In SP however, similar concepts can be requested to very different instruments and I believe that one of its interests is also to push the technical achievement of the performers to the limits of what they can produce with their instruments.

My motivation for this discussion comes from my experience of SP as a percussionist and the questions that I often see arising about how to perform a sign for a specific instrument, even though a definition of its prototype for the discipline has already been shown. From the previous point, one could think that a different concept is required per instrument, in order to perform a SP sign, just like there are different concepts per discipline. In the following part, I will reflect on the mechanisms that I use for achieving a LT on percussions and will argue that in each discipline, there is only one concept per SP sign, that does however require non-trivial technical approaches.

(2) A Long Tone on percussions

While the experienced performer will probably use different possible techniques intuitively (a fast roll, using brushes, playing on cymbals that have a long acoustic response), there are indices that the achievement of a LT is not as obvious on percussion that on a violin for instance:

- In the vocabulary of percussions, the term “long tone” is not used a lot or at all to describe a sound or its achievement. Instead, one would speak of a “roll” for fast, repetitive hits on the percussion, or brushing techniques.
- In SP context, I observed specific discussion between performers and soundpainters on the subject and soundpainters giving examples on how to achieve a LT with percussions, making a translation to “roll”.

[29] In this part, we won’t discuss the possibility that a concept has several prototypes, but the reasoning would be similar in that case.

In general, every concept does not necessarily have a trivial interpretation for all instruments. I interpret the fact that a "long tone" is not relevant to speak of a sound in percussions as the fact that there is not an "easy way" or basic technique that could refer to such a sound.

We have seen that in music, a LT is described by a prototype, whose characteristic traits include a constant volume and pitch over time. On a violin, we do not perceive micro excitations on a string but rather a constant acoustic response to the performer's long tone. By "percussion" however, it is meant that the acoustic response of the instrument is very sharp, such that listeners are usually able to discern the individual hits on the surface: our perception scheme will identify a note produced by drawing a bow with constant velocity on a violin as a long tone, whereas in general, it will not recognize the percussions sounds with naïve techniques as such. We also need to remind ourselves that sounds are not perceived as exactly a LT or exactly not a LT; we can apply the results of prototype theory presented in I.B.5 to conclude that people rather perceive them as "more or less" long tones.

Conceptually, the percussionist cannot perform the prototype of the LT but only "approach" it with several techniques: in the space of musical concepts as perceived by humans, the prototype of the LT appears the asymptotical, limit point of the concept of roll when its frequency goes to infinity. In other words, a very fast roll is perceived as "more a LT" than a slow one, and percussionists - just like every other musician - can play with the soft frontiers of the concept to respond to its request. The idea of technical and conceptual "approaches" to the prototypes of SP contents raises and explains other interesting features of SP:

- It is within the fuzzy limits of concepts that performers can explore new ideas, surprise the soundpainter and bring interesting material to the composition
- Soundpainters can push the performers out of what is usually called their "comfort zone" by requesting materials and interactions that are not trivial to achieve on specific instrument and favorizes imitation as a collective learning process
- Performers can find in the absence of clear boundaries the sense of freedom and improvisation that is necessary to them, as human artists. They can also subversively use it to disobey, contradict or overturn the soundpainter's requests.
- Just like there are "approaches" to a prototype, there are also "extensions" of concepts or detachment from their prototype. We will discuss in II.E.2 the role of cognitive load in such extensions of concepts during the learning of SP.

4. The Soundpainting grammar

Like other languages, SP has evolved with complex structures and rules that allow soundpainter to form sentences, i.e. to form meaning at a higher level than the signs themselves. Those structures are what define the grammar of SP, which allow soundpainters to communicate by creating temporal sequences of signs or combining several signs together. At the linguistic level, grammar can be split in two parts: morphology and syntax.

a) Morphology in Soundpainting

As defined in written and oral languages, morphology has to do with the internal economy of words. In English, a word like "*bookkeepers*" has four morphemes (*book*, *keep*, *-er*, *-s*) and is put together with morphology.

In sign languages in general, there are a variety of morphological systems and rule[30] that assemble basic signs (the equivalent or morphemes) in more complex ones (the equivalent of the words). Consider for instance the sign for "falling" in French sign language: starting from the icon of a person with one hand and the icon of the ground with the other hand, the movement of both hands represents the movement that a person would have when falling on the ground[31]. We can interpret this sign as a morphological construction from the morphemes "agent" (person) and "ground". Because of the relatively young age of several sign languages, their morphology is not always clear and shared among all speakers of the language.[32]

In SP, we can also observe several morphological constructions that historically explain particular formations of new signs. A controverted example in the SP community is the "change now" sign that is a morphological construction from the morpheme "change" and the morpheme "hit", borrowing the hand pose from the first and the movement of the hand from the latter one, creating a new sign that could be named "change now" (or "change hit"). It is also possible to think of the sign "glissando" as an

[30] Some basic examples can be found here: https://www.handspeak.com/learn/index.php?id=41

[31] See http://www.sematos.eu/lsf-p-tomber-5958.html

[32] For a discussion on this topic, see Aronoff, Mark et al. "The paradox of sign language morphology" *Language* vol. 81,2 (2005): 301-344. This argument is sometimes used to motivate the need for normalization of sign language.

extension of the morpheme “long tone” to the “imaginary ladder” that represents the mapping of pitches in the Y axis of the body (height).

These examples are raised in this report to point out the role of morphology in the grammar of SP. A deep study from linguists should be made to explore further its mechanisms.

b) Syntax in SP

(1) The way Soundpainting syntax is usually historically presented

In addition to morphology, syntax is the second important part of the grammar that defines how the different signs can be temporarily laid out to form sentences.

Historically, syntax in SP started to be discussed in the 90’s, long after its basic rules were already internalized and used by Thompson himself and the other few soundpainters that had learned the language (Minors & Thompson, s.d.):

> It wasn’t until 1997 during a Soundpainting residence in Woodstock, NY that I and Soundpainter Sarah Weaver formalized the syntax.

In his SP workbook 2 as well as on his website (Thompson, s.d.), Thompson presents the “structure of SP” in the following terms:

> The Soundpainting gestures are grouped in two basic categories: Sculpting gestures and Function signals.
>
> Sculpting gestures indicate What type of material and How it is to be performed and Function signals indicate Who performs and When to begin performing. Who, What, How, and When comprise the Soundpainting syntax. Note: The How gestures are not always employed. The Soundpainter often signs a phrase leaving out a How gesture. For example: Whole Group, Long Tone, Play. If you sign your phrase without a How gesture, then it is the performers choice in deciding the dynamics and quality of the material.
>
> The Soundpainting syntax Who, What, How, When and the two basic categories Sculpting Gestures and Function Signals are further broken down into six subcategories: Identifiers, Content, Modifiers, Go gestures, Modes, and Palettes.
>
> 1 – Identifiers are in the Function category and are Who gestures such as Whole Group, Woodwinds, Brass, Group 1, Rest of the Group, etc.

2 – Content gestures are in the Sculpting category and identify What type of material is to be performed such as Pointillism, Minimalism, Long Tone, Play Can't Play etc.

3 – Modifiers are in the Sculpting category and are How gestures such as Volume Fader and Tempo Fader.

4 – Go gestures are in the Function category and indicate When to enter or exit the composition and in some cases when to exit Content such as Snapshot or Launch Mode.

5 – Modes are in the Sculpting category and are Content gestures embodying specific performance parameters. Scanning, Point to Point, and Launch Mode are several examples of Modes.

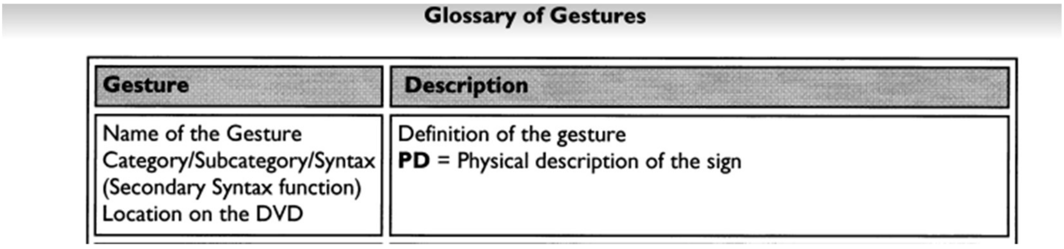
6 – Palettes are in the Sculpting category and are primarily Content gestures identifying composed and/or rehearsed material

(2) Initial remarks on the "structure of SP" presented by Thompson

We can see that by the term "structure", Thompson refers to two elements: the syntactic structures and the functions of SP (and perhaps the term "structure" could be refined in that sense; we have seen previously that there are other important structures in SP such as morphological structures). It is also in those terms that SP is presented during the workshops that I have experienced. Before going further, let's make some initial remarks on this introduction to the SP syntax:

- Walter describes a hierarchy of categories: the two initial categories "sculpting gestures" and "function signals" are "broken down" into the Who, What, How, When, Modes and Palettes sub-categories.
- The ordering at which the signs are sequenced in time is at the level of the sub-categories (Who, What, How, When – the ordering when using Modes or Palettes is not mentioned) but is independent of the meta-categories "Sculpting gestures" and "function signals". Consequently, the "Who, What, How, When" corresponds to the syntactic structure while these meta-categories are not syntax categorizes but my interpretation is that they rather represent Thompson's interpretation of their role in the sentence, from the point of view of the composer rather than the linguist.
- There are some ambiguities with the "Palettes" and "Modes" categories, that are said to also contain signs of the category "Content", although this division is at the same level.

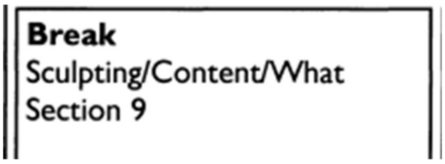
- On one hand, the “Who, What, How, When” categories can be seen as the analogies of subject, verb, direct object and other syntactic functions in written or oral languages we are familiar with. Moreover, these functions have a specific order that cannot be changed.
- On the other hand, the “Identifiers, Content, Modifier, Go gestures”, etc. seem to correspond to syntactic labels (also called syntactic tags) of the signs in SP.
- It is in these functional categories that signs are also described in Thompson’s SP workbooks 1 and 2.



Glossary of Gestures

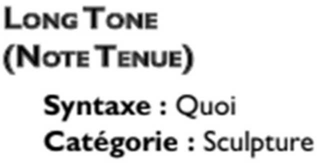
Gesture	**Description**
Name of the Gesture Category/Subcategory/Syntax (Secondary Syntax function) Location on the DVD	Definition of the gesture **PD** = Physical description of the sign

Figure 2 Structure of the glossary of gestures, WB 2 (Thompson 2009)



Break
Sculpting/Content/What
Section 9

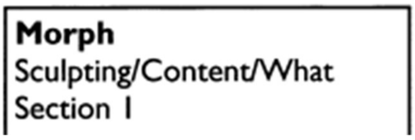
Figure 3 Name, Category/Sub-category/Syntax of the gesture Break in Thompson’s Workbook 2 (2009)



Long Tone
(Note Tenue)

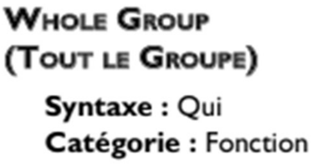
Syntaxe : Quoi
Catégorie : Sculpture

Figure 4 Name, Category/Sub-category/Syntax of the gesture Long Tone in Thompson’s Workbook 1 (2006)



Morph
Sculpting/Content/What
Section I

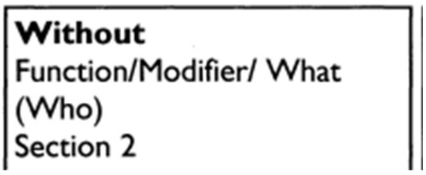
Figure 5 Name, Category/Sub-category/Syntax of the gesture Morph in Thompson’s Workbook 2 (2009)



Whole Group
(Tout le Groupe)

Syntaxe : Qui
Catégorie : Fonction

Figure 6 Name, Category/Sub-category/Syntax of the gesture Whole Group in Thompson’s Workbook 1 (2006)

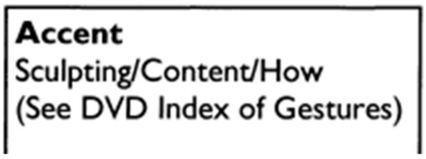


Without
Function/Modifier/ What
(Who)
Section 2

Figure 7 Name, Category/Sub-category/Syntax of the gesture Without in Thompson's Workbook 1 (2006)

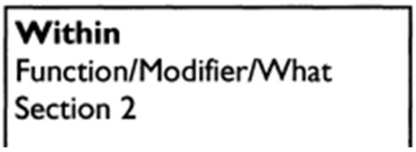
Although in the WB 1, the signs are only classified by "syntax" (Who, What, How, When) and "category" (function signals or sculpting gestures), in WB 2 the "subcategories" (Identifier, Content, Modifier, Go gesture) are introduced.

From the reading of the "structure of Soundpainting" and our initial remarks, we could conclude that there is an exact mapping between the syntactic functions Who, What, How, When and the syntactic labels "Identifier, Content, Modifier, Go gesture". However, we observe examples in the WB 2 that contradicts this hypothesis:



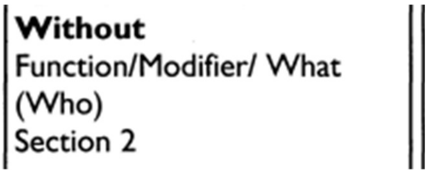
Accent
Sculpting/Content/How
(See DVD Index of Gestures)

Figure 8 Name, Category/Sub-category/Syntax of the gesture Accent in Thompson's Workbook 2 (2009)



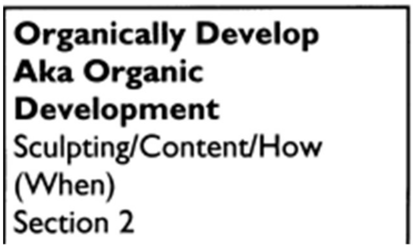
Within
Function/Modifier/What
Section 2

Figure 9 Name, Category/Sub-category/Syntax of the gesture Within in Thompson's Workbook 2 (2009)



Without
Function/Modifier/ What
(Who)
Section 2

Figure 10 Name, Category/Sub-category/Syntax of the gesture Without in Thompson's Workbook 2 (2009)



Organically Develop
Aka Organic
Development
Sculpting/Content/How
(When)
Section 2

Figure 11 Name, Category/Sub-category/Syntax of the gesture Organic Development in Thompson's Workbook 2 (2009)

In those examples, additional "syntax" categories are mentioned in parenthesis. We observe signs of the label "Content" which are classified as part of the "How" syntactic function and “Modifiers” that are classified as “What” or “Who”. Similarly, other signs are indicated as sharing the functions What and How, What and Who… and some Modifiers are classified as “When”. A priori, these could be seen as inconsistencies in Thompson’s analysis of the SP grammar. My interpretation is the following:

- First, I think that there is a confusion between the notions of lexical categories (and other syntactic structures) and functions in his analysis. It is suggested in his definition of the SP structure that each lexical category (Identifier, Content, Modifier, Go gesture…) corresponds to a single grammatical function (Who, What, How, When), while in his WB 2, he shows it can correspond to several functions. He does however not detail how a lexical category performs a specific function, or in which structures it can change from its “main function” to its “alternative” ones.
- Then, I believe that additional confusion is brought by the naming of his categories for the signs: “category”, “subcategory”, “syntax”. He misses two important things:
    1. What is named “syntax” corresponds to the typical functions that the sign can perform in the clause. However, it is different from the syntactic categories themselves (I suggest to name them “Identifier phrase”, “Content phrase”…), among which the lexical category is relevant for a single sign (Identifier, Content…).
    2. The “categories”, “Sculpture” and “Function signals” (at the opposite of the “subcategories” and “syntax”) may be relevant in artistic terms, in order to think the roles of the signs in the artistic creation (sculpting, function…) but are irrelevant at the syntax level.

If instead, he would rename those categories as “artistic category”, “lexical category” and “most common syntactic functions”, parts of the ambiguity would be removed.

- Moreover, I think that this hierarchy of categories has been designed to explain further complexity in the syntax itself, without studying its deeper mechanisms to derive the relevant syntactic categories in SP. In other words, by suggesting the existence of “alternative syntax” categories, Thompson is leaving implicit that:
    1. The syntax of SP has different hierarchical structures (phrases) that can be constructed out of several sign to perform a single function
    2. There are additional lexical categories such as adpositions that allow to construct phrases from single signs, while changing their function

As a simple example, let’s mention at this point that the common clause “minimalism group, off” is not explained in Thompson’s description, nor “Whole group, long tone with movement, play” or “Whole group, movement with long tone, play” (and it is unclear at this stage too which one of the latter is correct). In Thompson’s words (Duby, 2006):

> There are several gestures in Soundpainting which defy reasoning - irregular verbs so-to-speak. In other words, they break the rules of Soundpainting. A good example of this is the 'Watch Me' gesture, which gets used as an 'off' gesture when removing yourself from 'Shapeline' mode. Another example is the 'Synchronize' gesture in this case, the gesture isn't initiated by a 'Go' gesture - the content is performed once the 'Synchronization' gesture is signed. In most spoken languages there are irregular verbs the reasoning for their existence is confusing. I can't say how this came about in other languages, but I would venture to say it's probably for the same reason in the Soundpainting language, in each case, I had to break the rules to achieve my goals.

Unlike Thompson, I think that there are no gestures that “defy reasoning” in SP, otherwise they would not be taught, explained, understood nor used at all so commonly during SP workshops and rehearsals. My interpretation of his statement is that those gestures are not explained in his own vision of the SP syntax, by there are good chances that a deeper linguistic analysis would probably lead to more relevant categories to think signs with, in terms of their position in the clause, their function, but also their use in more complex syntactic structures.

- Finally, I think that his description suffers from the velocity at which the language has evolved from its most basic elements to complex uses and rules, leading to describe its structure both in artistic and linguistic terms and principles. This description might have been proposed at a time at which it made sense to think in those simple terms, while it was sufficient to introduce only superficial elements of syntax as the performers would internalize in practice many of the implicit complexity of the language.

The description of the syntax of SP would be the subject of a dedicated book, out of my field of expertise. However, I will underline some of its basic elements that will be relevant to my technical proposition of a recognition tool and propose a view on its mechanisms at several levels.

(3) A closer look at Soundpainting syntax

In this part, I will step back from the historical description of the SP syntax by Thompson to provide a wider view of its components from my own observations and interpretations.

(a) The big picture

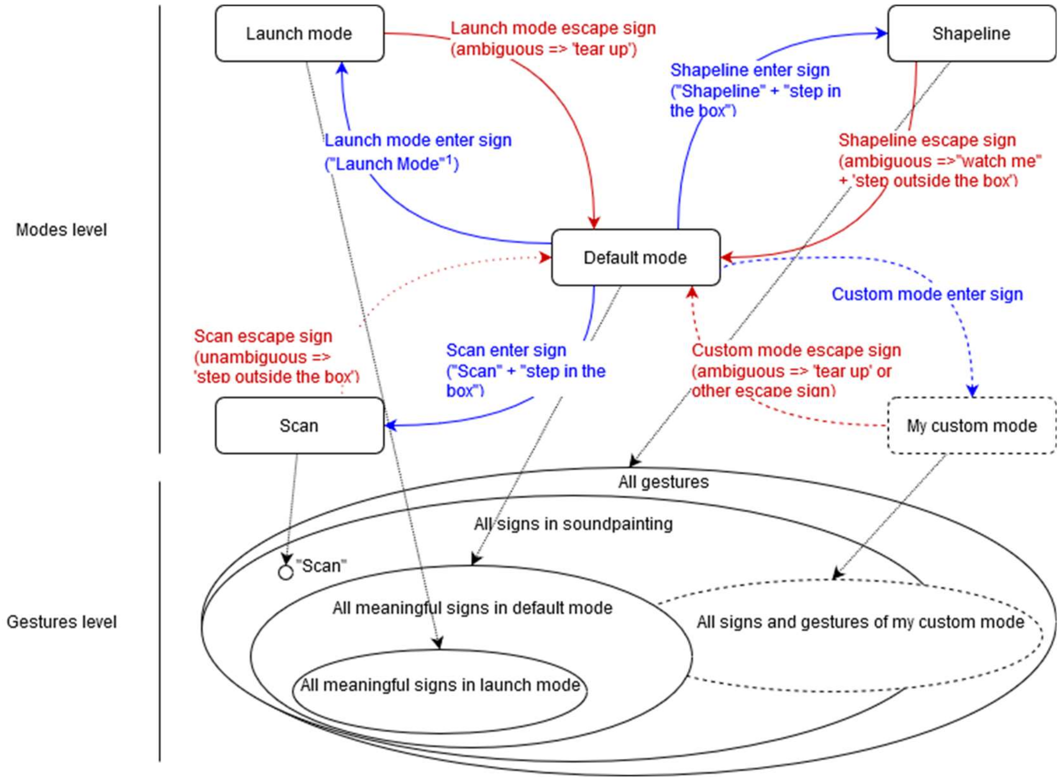

Launch mode
Launch mode escape sign
(ambiguous => 'tear up')
Shapeline
Shapeline enter sign
("Shapeline" + "step in the box")
Launch mode enter sign
("Launch Mode"1)
Shapeline escape sign
(ambiguous =>"watch me" + 'step outside the box')
Modes level
Default mode
Custom mode enter sign
Scan escape sign
(unambiguous => 'step outside the box')
Scan enter sign
("Scan" + "step in the box")
Custom mode escape sign
(ambiguous => 'tear up' or other escape sign)
Scan
My custom mode
All gestures
All signs in soundpainting
"Scan"
All meaningful signs in default mode
All signs and gestures of my custom mode
Gestures level
All meaningful signs in launch mode

Figure 12 Representations of the modes, their "enter"-"escape" dynamic with respect to the "Default mode" and the correlation of the ambiguity between two modes and their respective sign/gestures corpus: the "enter" and "escape" signs are used to disambiguate the interpretation of gestures or sign common to two or more modes for the performer. In the non-ambiguous cases, we see that the "escape" signs disappear (or is simply a "step outside the box").

[1]Launch mode is ambiguous because some signs must be interpreted as in shapeline (Identifiers for instance). However, it is not ambiguous for contents and modifiers, which can simply be performed immediately when the soundpainter "steps into the box". For that reason, their enter and exit signs are not always performed when there is no ambiguity. It is also possible that one particular mode is used by convention (by default) instead of the "default mode" (for instance, it is often the case that dancers interpret signs and gestures in launch mode during performance by default), in which case the enter and escape sign may not be used either.

Let's start with the big picture (Figure 12). Imagine a language that has an alphabet (a set of signs) and several syntaxes that all speakers know and can use whenever they think that one is more appropriate than the others. Let's call these syntaxes "modes". SP is such a language. We have seen that SP has an alphabet and how it is created. We have also seen that modes are said by Thompson to "embody specific performance parameters" (see I.C.2.b)(1)), even though they are at this point probably not usual Content gestures.

My proposition is to understand modes as several syntactic and semantic[33] modules that the soundpainter can use during the performance: they each encompass different ways of creating meaning from the association of signs (syntactic component) and may also change the meaning of several signs (semantic component). To understand what the common features between them are, how they relate to each other and are labeled under "SP", let's take a look at the alphabets of all these modes.

(b) A shared alphabet

Imagine the dictionary of SP and visualize the abstract space of all signs in SP (bottom of Figure 12). Among those signs, some are used in specific modes only. For instance, the sign "scan" is significant only for the mode with the same name, as it allows the soundpainter to enter the scan mode from the "Who, What, How, When"-like syntactic structure. In the following, we will call this syntactic structure the "default mode" of SP (the main reason for choosing this name (default) is that it is the syntactic module that soundpainters use by convention and which allows to connect all modes together). Outside the scan mode and its variations, the sign "scan" is practically not meaningful nor used. Therefore, when the soundpainter uses that sign, it is clear for the performer which mode the soundpainter uses.

However, some signs are significant and can be interpreted differently in several modes, which is represented in Figure 12 by the overlapping of two sets (ellipses) at the gestures level. When using those signs, it is no longer obvious what mode the soundpainter uses and how they should be interpreted: the mode is ambiguous, so there must be a way for the soundpainter to let the performers know what mode he is using. This is the role of what I call the "enter" and "escape"[34] signs for each mode.

(c) Ambiguities between modes resolved by the "enter" and "escape" signs

The enter and escape signs allow the soundpainter to "navigate" between the different modes, by signing the enter sign of one mode to "enter it" (i.e. use its syntactic and semantic structure) and signing its escape sign when escaping from that mode. However, not all modes in the graph are connected; in fact, there is a mode from which

[33] They embody several semantic components that can change the meaning of one or several signs and make some signs from the SP alphabet significant and some other insignificant.

[34] Walter uses the term « escape » in his description of the « tear up » sign which is very commonly used as an escape sign.

we can enter all other modes and that we return to by default when escaping a mode. I call it the “default mode” (or the “core”, “central” mode of SP), from which the other modes connect, just like modules in the system. Both the use of enter and escape sign is necessary when at the alphabetical level, the mode that is entered and the one that is left both make use of common signs. In that case,

It would be interesting to discuss further the motivations and consequences of this centrality of the default mode, and why it is not possible to directly move on from one mode to the others by simply signing the dedicated “enter” signs for each mode. If it was the case, I speculate that the common “tear up” sign would take the place of the enter sign for the default mode. I leave that discussion to further research and debates in the SP community.

(d) About syntactic differences and semantic similarities between modes

Syntactic differences are the most obvious ones to observe between modes. Whereas some modes have a structured syntax that allow to form complex requests (such as the Who, What, How, When syntax) or more simple ones such as “play can’t play”, some modes such as “shapeline” have no syntax: the performer must interpret freely, or in “abstract ways” the gestures and signs of the soundpainter, which can be any sign or gesture, even signs which are not part of SP directly. In general, we can classify the modes in two types that are relevant to the composer:

- Those that allow to create structured and delayed requests, in which the signs are not interpreted immediately but only at the end of a sequence
- Those that request the performers to respond to the signs in real time

To mark this polarity between “structured request and delayed response” and “immediate response to single signs or gestures”, the “imaginary box” is used. As presented by Thompson, the imaginary box is an imaginary space in front of the soundpainter that the soundpainter enters when he wants his request to be executed by the performers, opposed to the neutral position where the soundpainter prepares the request.

However, this description is only merely valid for the syntax of the default mode (Who, What, How, When -like): the imaginary box is also used in other modes, including those in which there are no structured requests. In fact, the box is a sign universal to all SP modes, with a slightly different meaning than what Thompson presents: by marking the end of the sentence, it indicates that the request must be executed. In the case of modes whose requests are not structured, all signs form a sentence by themselves and the soundpainter stays in the box when signing.

For all modes that are unambiguous with respect to the default mode, stepping out of the box is a sufficient sign for indicating that no more sentences are made within those modes. For instance, to exit the scan mode, it is enough to step out of the box and use the default mode syntax again, without an additional escape sign to signify the change in mode.

On the semantic side, many signs are shared by several modes and have similar, if not equivalent meanings in all these modes. In fact, the common semantic features of the different SP modes are what make them efficient and what allow their quick learning and memorization by the performers. Indeed, if instead there were only signs proper to each mode and new signs had to be learned for each one, SP would probably lose much of its performativity, which for me lies in its ability to mix several syntaxes with similar semantics elements.

(e) Remarks on the syntax of the "default" mode

We have seen that there is a diversity of modes, each encompassing their specific syntactic and semantic elements. In this part, we will to precise some notions of the grammar of the "default mode", which will be parsed by an automaton in my recognition tool. All the following discussion only refers to the grammar of the default mode.

(i) Lexicon categories

In linguistic terms, we can characterize each sign with a lexicon category that will be the lowest syntactic category in SP sentences. Those lexicon categories are the following:

- Neutral
  There is only one neutral sign in SP, which deserves its own lexical category as it can be understood as the analogy of silences between words in oral languages, and to some extent as the spaces between words in written languages. In terms of syntax, neutral can be signed at any moment in the construction of the sentence.
- Identifier
  The identifiers correspond to the signs that if not anticipated or followed by adpositions, will perform the function "Who".
- Content
  The contents correspond to the signs that if not anticipated or followed by adpositions, will perform the function "What".
- Modifier
  The modifiers correspond to the signs that perform the function "How" (irrespectively of their context).

- Timing
  The timings are commonly called “go gestures” and are the signs that perform the function “When” and mark the end of a clause (but not the end of the sentence).[35]
- Execution
  The execution signs are those that mark the end of a sentence when used simultaneously to a Timing or in conjunction with a Mode sign. In SP, the end of a sentence corresponds to the moment when the performers are asked to execute the request of the soundpainter. There is only one Execution sign, which is “stepping into the imaginary box”.
- Adpositions
  Adpositions are both prepositions and postpositions that cannot perform any function by themselves but create hierarchical syntactic structures when used in before or after Identifiers and Contents.
- Modes
  The Modes signs correspond to the “enter” signs of the different SP modes. When used in conjunction to an Execution, they perform the end of the sentence (from the perspective of the default mode only).

(ii) Syntactic functions, clauses and sentences

Similarly to what is usually introduced, the functions that are relevant to clauses are:

- Who
- What
- How
- When

Moreover, in sentences, we have the additional function Execution that is required to mark its end. Combined, they form the set of all functions in SP default mode.

(iii) Syntactic categories

A clause is a syntactic structure that comprises at least the functions Who and What.

[35] In the notion of “go gestures”, Thompson always assumed that it was the end of the sentence, whereas we can identify cases in which they are not.

A sentence is a conjunction of consecutive clauses terminated by an execution sign. Moreover, a sentence can only end with the simultaneous use of a timing and execution sign or the conjunction of a mode and execution sign.

An identifier phrase is a phrase that performs the function Who at the level of the clause. Similarly, a content, modifier or timing phrase are phrases that perform the function Who, What and When at the level of the clause.

(iv) Some production rules of SP grammar

We have seen previously the existence of several grammars that can be described as sets of production rules. I will give here an overview of the production rules of the default SP mode and discuss the category of grammar that it falls in. As quick notations, I use the operator of ordered conjunction +, the symbol "*" for every possible syntactic category, the symbol "Ø" for the empty phrase, the symbol "->" for "produces"; The set of non-terminal symbols start with an uppercase while the set of terminal symbols (the actual representant of each lexical category) start with lowercase. For the first sentence and later, there are the following production rules:

- * -> * + neutral
- Sentence -> Mode clause + execution
- Mode clause -> Identifier phrase + Mode phrase
- Mode phrase -> mode + Modifier phrase
- Sentence -> Clause + (timing, execution)
- Clause -> Clause + Timing phrase + Clause (clause duplication rule)
- Timing phrase -> Ø
- Timing phrase -> timing
- Clause -> Identifier phrase + Content phrase
- For all X in {Identifier phrase, Content phrase, Mode phrase, Modifier phrase}, X -> X + X
- Identifier phrase -> identifier
- Identifier phrase -> content + group
- Content phrase -> Content phrase + Modifier phrase (content's modifier extension)
- Modifier phrase -> preposition + content
- Modifier phrase -> Ø
- Modifier phrase -> modifier

So far, it looks like a context-free grammar and does satisfy the corresponding pumping lemma: indeed, to construct an arbitrary long sentence, one must necessarily use either an arbitrary number of clauses duplication rules or content's modifier extension rule. However, there are additional rules that introduce a sensitivity to the context, and that are better expressed with words:

- One identifier cannot be used in conjunction with a content in two different clauses, otherwise it would be ambiguous what content is requested.
- After the first sentence, the grammar changes, allowing for omitting one or two functions between (Who, What, How). When a function is omitted, the clause implicitly refers to the last sentence in which it was provided.
- Some identifiers are also themselves context sensitive, such as “Rest of the group”, which implies a knowledge of the identified performers in the previous clauses or sentences.

My understanding is that these additional rules imply that SP is a context-sensitive grammar. However, I cannot prove it at the moment of this report, and I suggest that the question is treated in further analyses of SP, with the identification of all production rules of the grammar. Whatever category the grammar falls in, it still allows for parsing from an automaton, which I will implement in Max/MSP.

(v) Specificities and cultural inscription

The “default mode” appears as a very unique element of the SP syntax because of its centrality in the network of modes (Figure 12) and its use as the conventional syntax of SP, to the extent that SP is presented in its grammatical terms, despite the number of other syntaxes it covers. To my knowledge, it is also the only mode that incorporates syntactic elements and rules from Western oral and written languages. Those elements, such as recursion, omission, prepositions (both simple and complex adpositions), postpositions or logic operators allow for greater complexity in the request structure and are usually implicitly learned (i.e. they are not explicated by the teacher or soundpainter).

In linguistic literature, we have observed several languages that do not possess these concepts.[36] One could wonder how much presenting the syntax of the default SP mode to communities that are not familiar to Western languages would challenge the usual conceptions of the SP syntax. The story of the emergence of the Launch Mode (Minors & Thompson, 2012) is an answer to that question:

> Launch Mode means to respond to the What gesture immediately – a Go gesture is not needed. In other words, when the Soundpainter signs a Long

[36] The most interesting example to my mind is that of the Pirahã people studied by Daniel Everett, which contradicts the universality of recursivity in human cognition.

> Tone, at the moment of signing the gesture, the group responds and performs a Long Tone.
>
> The idea for this gesture came from my work with the learning impaired. It isn't always possible for people with learning disabilities to comprehend a Soundpainting phrase incorporating the syntax in its entirety – Whole Group, Long Tone, Volume Fader (ppp), Play. This type of phrase wasn't possible to follow the first time I Soundpainted with a group of learning impaired people. So I ask the group to respond to the Long Tone gesture at the moment I signed them. I gave an oral example of what was expected and the group found it very easy to comprehend. I took the same approach with all the other gestures I taught the group.
>
> Afterwards, I decided to use the immediate response to a What gesture and created Launch Mode. Nowadays, Launch Mode is a widely used gesture amongst many Soundpainters and its origin came from necessity.

In this example, we can see that the default mode does have a degree of complexity that involves prior knowledge in grammatical rules and logic. Of course, all mechanisms of language are not cultural and learned, but these rules were not internalized by the learning impaired and it might as well be an issue for working with people whose language or culture does not yield them either.

(f) Conclusion

I suggest for future research to push the analysis of the SP syntax and its formalization in linguistic terms. At the moment I am writing this report, a dictionary is being built that already contains some descriptions of signs in terms of syntax, such as the ones that have historically been formalized by Thompson. I believe that a linguistic approach to SP would be very relevant and add a lot of value to this collaborative work.

D. Meta-structures of Soundpainting: communication outside sign language, configurations, conventions and diffusion mechanisms

In the previous sections of this report, I have discussed several mechanisms of SP such as its linguistic and intrinsic structural features. In this section, I would like to first point out that it cannot only be described in those terms and limited to those mechanisms. To do so, I will give an overview of some extrinsic aspects of SP that play important roles in the way that it is used and understood by those who practice it and discuss the implications of the linguistic modeling of SP in a critique of some of the traditional definitions and presentations of SP.

1. Preliminary remark: what we mean when we refer to "Soundpainting"

I observe that what is usually called "SP'' is not only the communication between the soundpainters and the performers with signs. The term also covers the whole context and performance configuration that surrounds it and establishes a broader communication between the soundpainters, performers or the public. It is important to clarify that the name "SP" is commonly used to signify not only a sign language but an artistic practice as a whole: a context, configuration, esthetic, conventions and several types of communication… While I won't enter the process of re-defining SP, I would like to discuss in the following parts what I call the "meta-structures" of SP, which are all the structures and mechanisms above the description of SP as mere sign language that compose its usual definition.

2. Soundpainting as a complex communication form

We have seen that SP allow for a communication from one or several senders (the soundpainters) to receivers (the performers): so far, we have only discussed SP as a linear communication form using sign language. In general, we can identify multiple interactions within a SP group, including feedback to the soundpainter's request. The soundpainters may receive several forms of feedback:

- The artistic productions of the performers
- The unintentional communications of both the performer and the public
- The few signs that can be used by performers as such (not as soundpainters)

Although there are some configurations in which these elements of communication do not occur, soundpainters are usually able to perceive feedback to their requests. One may object that they can be prepared in advance and written down on some kind of score, such that the feedback from the performers would have no influence on the linear communication from the soundpainter to the performers. It is however not the case in general and it is often reported and discussed how the requests are made in reaction to the proposals of the performers. SP as a "real time composition" implies the role of the soundpainter as a composer who listens to and shapes the artistic propositions of the group.

3. Configurations of Soundpainting practices and performances

a) Introduction to the notion of configurations and contexts of Soundpainting performances

In this part, I will raise show some contextual features that play an important role in SP and point out the historical emergence of new configurations in which conventions that are usually presented (by Thompson for instance) as representative of SP such as the traditional dichotomy between the composer and the performers (or the orchestra) disappear.

We have seen that SP is introduced as a sign language between a soundpainter which is the composer and an orchestra that is the set of performers. We also have seen that these roles are questioned and that SP is used in different ways by dance duets, jazz bands and other groups in which performers also sounpaint, or soundpainters are also seen as performers themselves, while the composition process is not be directed by a composer anymore, but may be a collective process in which the notion of composer makes no or little sense.

I would like to call these different "ways" of using Soundpainting and their contexts *configurations*. A configuration may embody many parameters such as the distribution of the roles in the group, what are the possible interactions with the public, spatial arrangement, cultural conventions, particular compositional rules and other elements of context that are not defined using the sign language itself. I use it to mean in general "the context of the performance and how people decide to use SP within the performance".

(1) Remark on the sign "configuration"

There is a sign in SP called "configuration", whose description in the WB 2 is the following:

> Players maintain the parameters of an assigned role, individually, or in a specified group or groups. A Configuration may comprise style and/or specific role function for the performer. Configurations are usually assigned in rehearsal. A Configuration may have a style assignment such as a performers role commonly found in most traditional jazz ensembles.

Such a definition does relate to notions of roles and compositional conventions. However, my own use of the word “configuration” is perhaps broader than its meaning in the SP language and does not directly refer to its use within the sign language.

b) Linguistic re-definition of SP taken away from its traditional configurations and contexts

At this point, it is important to motivate the concept of configuration by stepping back from the historical introduction of SP by Walter Thompson and reflect on what SP represents today and might represent it in the future.

(1) Performativity and choice of a language

Let’s now assume that SP is a mere sign language, that is only defined by its grammatical and semantic features. One may use this language for cooking, buying clothes on the market or playing games. In all of these uses cases, SP may seem very inefficient for communicating relevant messages and meanings. However, it might be performative in a band, in a group of dancers or between a composer or a conductor and an orchestra.

The choice of a language is motivated by its performativity in a context, while it can be used in other situations: the language is not defined by it.

(2) Configurations as external elements to languages

SP may then simply be understood as a language among others, used in specific configurations for which it is more convenient than others but not defined by its use in cooking, playing games nor real-time compositions. To my point of view, the fact that it is mostly practiced between composers and orchestras is a result of its history, diffusion, adoption processes and what it has been designed for and performative at but is not a constitutive element of its definition. Of course, what we usually mean by “SP” does not refer to it as a mere language and also relates to elements of context such as the configurations it is used in; but my point here is to argue that configurations are not constitutive elements of languages, while it is true that languages are used in specific configurations.

We can make an analogy here with oral or written languages, that are convenient for cooking or playing games but perhaps not as good as SP for real-time composition with a multi-disciplinary orchestra. Yet these languages, such as English, are used in a variety of configurations: from a collective communication between several family members in their house in England where it is born to a monologue of a foreign indigenous tribe representative at ONU. English can no longer be defined as the language of families in England, nor as the international language for ONU speeches in 2020; it is also used in many other contexts and configurations. In fact, we know from psychology and from the studies of concepts that there may not be one definition of English at all but only several perspectives that combine into a single complex concept.

(3) Critique of compositional choices and configurational choices as definitions and rules of Soundpainter

Similarly to my previous example with the English language, I would like to push for a similar understanding of SP as a complex concept involving a sign language and set of practices that are not defined by any configuration or context. For instance, let's consider the following statements, that are found in Thompson's presentations and definitions of SP:

- "The soundpainter is the composer"
- "The gestures are signed by the soundpainter" (here, I would like to point out the unicity implied by "the")
- "The Soundpainter, standing in front of the group (usually), signs a phrase to the group then composes with the responses."
- "One of the most important aspects of Soundpainting is to compose with what happens in the moment whether it is intended or not"
- "A very important part of Soundpainting language is the basic rule that there is no such thing as a mistake"
- "It is the Soundpainter's responsibility to realize the piece"
- "[The soundpainter is] the creator of the work, the director of the performance, the instigation for communication among an ensemble, the owner of the work"
- "All performers must keep the Soundpainter in their vision whether directly or in their periphery. This must be so in order for the Soundpainter to communicate the next phrase and be able to continue with the development of the piece."

I propose that these statements are not considered as properties of SP nor elements of its definition but as either configuration or compositional choices - that of course are important parts of SP but not constitutive of its definition. To demonstrate this proposition, I would simply draw attention to the examples of section I.A.2.c) above which contradict those statements. Indeed, we observe in such "revisited" orchestras that:

- Performers can soundpaint while performing, hence creating a shared composition and voiding the notion of composer (and unique composer) in the definition or as a property of SP
- Several compositional rules that Thompson defines as SP rules are changed or not considered at all:
    1. Some soundpainters do not require all performers to always keep them in vision.
    2. Some soundpainters may as well consider that it is not an error to ask for new content or mode without first indicating what to for with the material being performed, and may as well consider by default that the performers must stop their previous artistic production when it happens
    3. Some groups do not consider any responsibility of one or several soundpainter to realize the piece, promoting more democratic structures where each has an equal authority and responsibility in making compositional choices, performing and realizing the piece.

       All these points are also relations of power and hierarchy between the soundpainter and the performers. It is important to note that configurations also embody power and hierarchical relations, that may appear also in the form of spatial configurations (the soundpainter sountpainting in front of the orchestra, following the traditional model of the conductor).
- SP is used in configurations where it is not the only or most important language - for instance within a band to communicate during a performance -, hence not having a great influence or an influence at all on the philosophy of the composition or artistic production.
- SP is used outside the artistic field and in situations where the notion of orchestra, composer, discipline, etc, are irrelevant to some extent. For instance, SP has been used to control a swarm of autonomous drones (REF), by directing the movements of several group entities. In that case, the drones can be thought of as performers who also share a form of communication (GPS coordinates, machine communication protocols) and with whom the pilot (the soundpainter) is requesting a number of things. My proposition of recognition is another configuration in which many of those statements cannot apply either, whereas I consider that it is also SP.

At the opposite of these new configurations, we can read the vision of Thompson on SP in the compositional rules, configurations and contexts that he defines it with and in which it has grown. However, they may now not be used anymore to describe and define all the varieties of contexts and configurations in which it is used, nor all alternative compositional choices and "philosophies" that are brought with them.

c) Default parameters as conventions and configurational elements of Soundpainting

(1) Confusion between default parameters, conventions about the interpretation of sentences and grammatical rules in Thompson's descriptions of SP

In his WB 2, Thompson introduces the complex concept of layer in SP with the following sentence:

> There is a general rule in SP which states: Whenever any Content is being performed it is an error if the Soundpainter uses Scanning, Point to Point, or any of the other Modes or Content gestures without first indicating what to for with the material being performed.

He then explains how the concept of layer solves this problem by requesting the performers to play those modes or gestures "above" the layer of original material and to return to it when the Scanning, Point to Point or other Content gestures do not require them to perform. I would like to raise the following points:

- In his description, the error is a grammatical error (the sentence is wrong), hence the "general rule" is in his words a grammatical rule of SP.
- Part of Thompson's statement is inconsistent with very common similar situations in SP that are not considered as errors. In his statement, Thompson describes the ambiguity on the side of the performer of being requested a new content, for instance with Content gestures without first indicating what to do with the material being performed. However, in practice, this situation is observed very commonly and even in the compositions and teachings of Thompson himself. Take the following basic example: "whole group, long tone, play. Whole group, minimalism, play". According to his statement, the second phrase is an error because the performers have not been indicated what to do with the ongoing long tone. In fact, we learn by convention in SP that the long tone must be stopped by the performers at the moment that the minimalism is requested by the soundpainter; I have never seen a soundpainter signing "[step out the box] whole group, off, minimalism, play [step in the box]" to tell the performers to stop prior content before performing the minimalism.

The fact that a prior content must be stopped when a new one is requested is a convention that has been formalized in SP as "default parameters" (Peluci de Castro, 2015):

> A Default may be a rule either set in rehearsal, or during performance, which gives the Soundpainter a basis from which to compose such as Actors speak without gesturing (only use their voice) or Dancers remain seated unless otherwise signed to stand up and move in the space. It is very important when working with Actors and Dancers to set your desired Defaults so the performers will know how to respond in certain situations – what are the parameters they perform within. If you have not set your Defaults then you and your performers will most likely experience problems during rehearsals and performance.

If one decides to ask the performers to behave differently, the new parameters can be conventionally defined prior to the performance and assigned to specific signs and indexes[37]. In his statement, Thompson seems to forget that the situation he describes is very common but is in practice never judged as an error, because of the conventions that are learned on how to interpret such ambiguous requests.

- At this point, one could still argue that even though the error is conventional and not grammatical, it is still a rule of SP.

We have seen in I.D.3.b)(3) that in the conventions of SP, the mistake has no place on the side of the performer. At the exception of grammatical errors (which in this case it is not), I then wonder why would mistakes exist on the side of the soundpainter. Indeed, Thompson justifies the inexistence of mistakes on the performer side by the interesting propositions and situations it creates for the composition.

I believe that in general, mistakes on the side of the soundpainter could also result in interesting compositional propositions, ambiguities and unexpected behaviors that are the core of serendipity. It is well known that languages develop some mistakes and that they are most of the time the result of unconscious analogies (Hofstadter, 2009) (and possibly other mechanisms) that reveal the role of hidden structures in the understanding and production of language. In fact, most of the language we use for our everyday life comes from such mistakes and uncontrolled evolutions of the grammar.

- Finally, I propose to interpret his statement as a mark of his own conventions as a composer and see in such a rule the particularity of a configuration rather than a

[37] Without going into many details, the concept here is simple to assign each set of parameters to a number (an index) to be able to change them during the performance.

"general" syntactic rule. It is obvious that most of the structures and conventions in SP follow closely the visions of Thompson on composition and what he does expect from the performers himself.

Indeed, a different composer could as well propose by convention that scan, point to point and all content gestures are meant as additional layers on the ongoing material. Such conventions are what allow performers to interpret ambiguous requests in other situations (cf. the example in the first point); but it may be the case that Thompson or other soundpainters disagree on the conventions to use when the soundpainter indicates new modes.

As a composer Thompson developed and formalized an important set of conventions from his own experience (we can think of the conventions about the development rates, neutral positions, or simple conventions such as what defines a minimalism or a pointillism) that he diffuses as a main figure and teacher of SP.

This example also points out the important role of conventions and other sets of rules that are not explained from within the grammar of SP but are determined by the configuration of the group, for instance by a composer.

(2) Remark on the diffusion of conventions in Soundpainting

The previous example underlines the conjunction of two roles of Thompson, first as a composer and second as the main diffuser and authority of SP, who defined SP rules not only in terms of grammar and semantics but also in terms of conventions such as default parameters and configuration choices.

In the previous parts (see I.D.3.b), I have tried to push for a conceptual separation between SP as a sign language and the elements of context such as conventions and configurations that come with SP performances. This separation is analogical to language and culture: the culture defines the use, conventions and configurations of a language; the language itself can be described in mere linguistic terms but cannot be understood without elements of culture, that are however not necessarily elements of language. I believe that there is an implicit confusion in Thompson's descriptions between the compositional conventions in SP and its definition of SP as a sign language. I explain this confusion by the following:

- The most obvious point is that Thompson is not a linguist himself and started to think about SP in linguistic terms dozens of years after internalizing his own representations of SP in artistic and compositional terms.

- He is historically the creator and main diffuser of SP and teaches his compositional conventions and propositions as part of the language itself; he presents SP as a composition tool and what he designed and developed it for, not only as a mere language.
- SP may be most appreciated and adopted for the set of compositional propositions and concepts from several disciplines that it encapsulates rather than its internal mechanisms as a language such as the creation of new signs, modes and syntactic structures. In other words, the adopting of SP may rely more on the exploration of its different modes, contents and "built-in" categories than the creation of new modes, contents, signs and structures.

We have seen that it indeed borrows ideas and artistic concepts from several artistic disciplines and provides a convenient context to explore them, which makes it a very powerful tool to move on from other configurations of improvisation, real-time composition or experimental music.

- Finally, the "traditional" conventions in SP might be related to a Western vision on arts, performance and their cultural conventions. The fact that SP developed a lot in Europe may have had an important influence on the culturally determined configurations it was used in and the people who diffused it.

I interpreted those specificities in the mechanisms and contexts of diffusion of SP as explanations of what I would call rather mainstream or traditional configurations of SP in contrast to marginal, recent configurations.

d) Remark on the performativity of Soundpainting, style and esthetics

Walter Thompson writes in his WB 2: "At this point in your studies you understand Soundpainting is not a style of music such as Classical, Jazz, Rock, Rap, Folk, etc. It is a sign language for composing in real time". What is raised here is the fact that one can use Soundpainting for very different esthetic and stylistic results. However, I will try to discuss the practical extent of the esthetics and style productions with SP, that are in practice very oriented towards certain types of music only.

On one side, SP is not equally used to produce all styles and esthetics. SP is more efficient and performative at composing in certain styles than others, which may explain why some artists are more interested in SP and some are less. This also explains why some people say that something "sounds" or "looks" like SP, as they do construct a prototypical view of the style of SP from their experience.

On the other side, the production of a certain style or esthetic, the choices of conventions and configurations in SP are conditioned by both particular (individual) and cultural contexts, which may also have influenced its development towards a greater performativity for certain styles and esthetics, for instance linked to the Western contemporary artistic practices.

Because SP is now developing in many countries, cultures and is being used by an increasing number of artists, I am sure that the diversity of its use will keep increasing, from experimental contemporary performances to baila baila dancing jam sessions.[38]

4. The meaning of Soundpainting as its use

In the previous parts, I have been critical about the deploying of many elements of contexts inside the common definitions and descriptions of SP. However, in this part, I would like to step back from this critique and reflect on SP from the point of the Austrian mathematician and philosopher Wittgenstein for whom the meaning can only be a function and definition of how the language is used in very practical situations.

In accordance with what we have seen of the concept theories, Wittgenstein reminds us that definitions are constructed from the common use of words and language, rather than linguistic analysis and theoretical models. The different definitions of what is a language (see I.B.1.d)) is symptom of the different perspectives and contexts in which this concept is mobilized: some consider English from a linguistic perspective, while some see it as a *langue* and a *parole* from a particular culture, territory or history. Both definitions are at the same time valid and bring elements of understanding to what English is. The same applies in SP. Although I am myself influenced by theoretical models and approaches that do not and cannot account for all the extent of the concept of SP, its contexts of practice, history, conventions or cultural aspects, I can only invite the reader to consider such elements as at least important to understand SP in both its traditional and new practices, if not what constitute the meaning of SP precisely.

In this broader frame, the contribution of linguistics is in my perspective not to normalize a language, decide how it should legitimately evolve, nor be practiced and used. I think that history have shown us how tempting this change of paradigm is for those who are able to describe and think a language in grammatical, structural terms,

[38] I am referring here to the concerts/parties that are held by the Soundpainting group of Rio that I had the honor to meet and play with, in contrast to my SP experience in Europe.

deriving what we are taught in schools as “laws”, forgetting their conventionality, futility in time and diversities in cultures or even inside the group of speakers itself. In France where I grew up, I could observe and read about the deeper sources and implications of normative institutions for the creation of a “society”, the reproduction of established forms of power and segregation. At the level of SP, these ideas are not insignificant nor irrelevant to me. There are also established forms of power, legitimacy and normalization at a certain point. As Foucault remarks, they are not localized but rather to be expressed in several places, diffuse and shared among people inside the SP community for instance. But at the moment that a dictionary is being built, that my own research tries to lead to better understandings of the mechanics of SP and that the figure of Thompson still play an important role in its diffusion and appropriation, I feel the personal need to open myself to the its marginal, limit, new, divergent or “alternative” practices and document them with the same interest as what is more univocally and commonly seen as SP.

E. Conclusion of the theoretical part

In this theoretical part, we have been able to frame Soundpainting inside broader historical and theoretical contexts.

We have seen that Soundpainting builds on elements of communication through gestures that have developed in millenniums and spatial representations of the qualities of sound and movements called *metageometries*, that are particular to the Western tradition. It is however symptomatic of the research for new means of expression, compositional techniques and languages of the XXth century, and can be related to other propositions of the same epoch such as Conduction or artistic practices within the deaf communities.

I then provided a broad constructivist model to SP by identifying in a bottom-top approach the general structures of SP in linguistic terms. I showed at the lowest level mechanism of creation or borrowings of signs, which can be described as a transformation from concepts of different sources to several types of signs performed with gestures. I have identified these types to Peirce’s triadic classification of signs and shown that *metageometries* appear in the morphology of SP faders.

At the second level, I discussed what I called the “overloading” of signs, that consists in both the polysemy of multidisciplinary signs across disciplines and the operability of each signified concept among instruments of the same discipline. I described the polysemy of multidisciplinary signs as a historical construction relying on cultural analogies between different concepts, each belonging to one discipline. Moreover, I pointed out the essential difference between disciplines and instruments by showing that at the contrary of disciplines, instruments of the same discipline share the same concept of SP signs, that they perform however in different and sometimes non-obvious technical approaches.

At the highest structural level, I criticized the common descriptions of the grammar of SP and pushed for a deeper analysis of its syntax. I suggested that "modes" are seen as several syntactic components that are linked by a shared dictionary and a central grammar that I have called the "default mode". Moreover, I unveiled some ambiguities in Thompson's classification of signs and provided a finer view on the production rules of the default mode grammar.

Finally, I discussed the meta-structures of SP from my point of view as a performer, soundpainter and observer to show that it could not be understood outside its elements of context and the complexity of the broader communication within the group and eventually the public. After raising several conflicts between the "traditional" descriptions and definitions of SP of Thompson with what I consider marginal and recent practices of the language, I could point out to mechanisms of authority, legitimacy, diffusion or appropriation of the language that play important roles in the identification and normalization of Soundpainting.

This theoretical exploration is a basis for the practical part that is following. However, many elements of theory appeared from the confrontation of my previous experience to the construction of a digital tool which requires specific implementation choices and the translation of implicit and internalized knowledge about SP. Both theoretical and practical paradigms are evolving one with the other; I consider my contribution to the understanding of this language as a work in progress.

II. Practical part: Soundpainting recognition with Max/MSP

In this section, I introduce my Max/MSP Soundpainting recognition tool. First, I explain my motivations for the creation of such a tool capable of recognizing several SP signs but also of defining new ones, as a composer would do. Then, I explore its structure in a top-bottom approach, in contrast with the constructivist model of SP presented in previous sections: I present the global structure of the program before taking a deeper look into the features and key objects of each layer. Moreover, I investigate basic performance mechanisms to optimize the speed and accuracy of the system. Finally, I discuss the use of this tool for learning SP in connection with the theoretical aspects discussed in the previous parts and look at some implications of the tool for the formalization of SP and hopefully, the future of the language.

A. New configuration: motivations, goals, workflow & challenges

1. Motivations

a) Computer music with Soundpainting

In his interview by Minors (Minors & Thompson, 2012) Thompson reports the specificities of using and controlling electronic instruments with SP:

> [Helen] Electronic are incorporated into your work and gestural language. What software have your laptop performers worked with in the past?
>
> [Thompson] Each laptop performer usually designs or modifies their own software in order to be able to respond to the gestures quickly.
>
> [Helen] Have you found any limitation in the real-time nature of the response from these electronic ensemble members?
>
> [Thompson] There are times during a Soundpainting where I have signed a laptop player to perform a phrase and they are not ready to do so. This is common and each Soundpainter must set specific Defaults during rehearsal in order to deal with this problem. For example: If a laptop player is signed a gesture such as Pointillism, but they are not ready to perform it because of their software or any other reason, they must sign to the Soundpainter the gesture called "I Can't Do This". The Soundpainter will either wait and sign the same phrase a little later in the piece, when the laptop player is ready, or will discard the idea and go on to another. This is a commonly used Default when working with any discipline that may need a little extra time to prepare a response. This is the same when working with visual artists – they often need a little more time to prepare a response to certain gestures.

He points out the difficulties that performers using electronic instruments can have when being requested immediate contents. Electronic instruments have sometimes interfaces that allow for complex parametrization of the instrument at the expense of the time of accessibility of the features.

Using gestures to control these electronic devices directly would bypass their standard interface for providing real-time control to the instrument in the context of SP: instead of changing the parameters of the device in real-time using the interface, one will be able to build before the performance a mapping between signs and parameters (or presets) and change those in real time. The binome (sign language, mapping) can therefore be considered as building a new interface to these devices that fit the needs of performers and composers.

b) New configurations

We have seen previously (see I.D.3) that SP is used in different configurations. Each configuration acts as a super-structure in defining its set of internal grammars (for instance, the modes), the emitters/receivers of the sign language (soundpainter/performer) and the “default parameters” as context-depend rules for the performance.

By creating a SP recognition tool, I am proposing new configurations for SP, in which a computer can process SP signs and take the role of one or several performers at the same time, aside from other human performers or not. At the difference of what is possible with human performers, the program itself can only recognize signs from one sender at the time (the soundpainter) and cannot send SP signs itself (but it can of course provide other types of feedback). Among the set of possible configurations, I would like to mention two of them that I had in mind when starting this project:

1. In the first configuration, the orchestra is simulated by the computer running the SP recognition tool (and possibly other digital gear), without other human performers.
2. In the second configuration, the computer takes the role of single performer inside an orchestra with other human performers.

In general, the choice and use of a specific configuration is motivated by the qualities (in terms of creative processes, special layout, communication rules…) it has for achieving a certain result in the performance. Let’s discuss the qualities of these two configurations:

- In the configuration one, the tool can be used for learning by individuals aside the collective approaches to learning SP. As for now, the tool only covers the basic structures of SP and would be interesting for beginners in SP or soundpainters who cannot rehearse with a group.
- In both configurations, it is interesting - even for more advanced soundpainters - for exploring new areas of composition with artificial intelligence, machine learning and their complex generative processes.
- In the second configuration, the tool can be used as a controller with other digital elements that are already part of the performance and sometimes used by the human performers themselves: effect processors, amplification mechanisms, mixing devices, recording devices and much more. The interfacing possibilities

offered by Max/MSP makes it an ideal choice for controlling these devices within the SP language directly and explore computer music.

Of course, there are other possible configurations in which people would want to use the tool, such as remote performances over the internet. In that case, the tool could act as bandwidth reduction mechanism by transforming video information (heavy) into sets of positional features or sign labels (very lightweight).

2. Goals

In the frame of my master thesis, I have chosen to focus on the use case of my tool as a learning tool in the first configuration (simulation of the whole SP orchestra). In analogy with the constructivist model discussed previously, the learning tool must have at least the following features:

- A mechanism of sign & dictionary creation
- A mechanism of classification between different signs and parametrization of different positions in the body space
- One or several grammatical systems for parsing the sequence of signs and gestures (one grammar for each mode)
- Audio/visual feedback in response to the soundpainter's requests

The goal of my project is to implement these features, focusing on only one mode (the default SP mode) and implementing audio feedback by simulating a small orchestra.

3. Music through movements: influential projects and existing tools

At the time I started thinking about this SP recognition tool, I was influenced by several other projects on the topic of music synthesis through movement:

- The HEM (Haute Ecole de Musique) of Geneva is developing with IRCAM (Institute for Research and Coordination in Acoustics/Music) a tool named GeKiPe (Geste Kinect et Percussion)[39], a gesture-based interface for audiovisual performance by Philippe Spiesser, Thomas Penanguer, José-Miguel Fernandez and Alexander Vert. Watch it in action or in this more explanatory video.

[39] See http://ensembleflashback.fr/transmission/gekipe/

- The MIMU gloves[40] were inspired by SP[41] and are one of the only commercially available products that let artists control devices and sounds in real-time from the movements of their hands. They allow for both assigning poses and movements of the hands in space to parameters such as effects and to triggers for events such as sounds or beats.
- The V Motion Project is a project directed by Jonny Kofoed & Matt von Trott, Assembly Ltd. in 2012 that uses two Kinects to build a gamified playback of a song in the same fashion as the MIMU gloves system, by triggering events and controlling basic parameters. They made an impressive show for a commercial clip with a very attractive interface which is probably the greatest value in the project.[42]
- At IRCAM, The Sound Music Movement Interaction team has created a set of probabilistic models as part of the MuBu library for Max/MSP and use it with several artists for creations linking movement, music and computing.[43]

c) Literature on Soundpainting recognition

Prior to my project, there have already two series of attempts at SP recognition:

- In 2014, the article "Towards Soundpainting gesture recognition" was the first one to my knowledge to propose a proof of concept with the recognition of a few SP gestures from Kinect input with Hidden Markov Models (Pellegrini, Guyot, Angles, Mollaret, & Mangou, 2014). The same authors wrote a second article in 2016, « Vers la transcription automatique de gestes du soundpainting pour l'analyse de performances interactives » (Guyot & Pellegrini, 2016), focused on the speculative use case of their recognition system for the annotation and the reconstruction of the gestures that were made during the performance. However, they only discuss the possibility of annotations and other use cases of the recognition system theoretically without proposing a concrete prototype for it.
- In 2017, Couture wrote "Using the Soundpainting Language to Fly a Swarm of Drones" (Couture, Bottecchia, Chaumette, Cecconello, & Rekalde, 2017) in which she uses a Kinect to pilot drones, each content being assigned to a specific drone task. Later in 2019, after a follow-up on the use of SP with drones, Couture,

[40] See https://mimugloves.com/

[41] I learned it in a conversation with their developers.

[42] See the clip at https://www.youtube.com/watch?v=YERtJ-5wIhM.

[43] See https://ismm.ircam.fr/.

Jáuregui and Dongo released another article, this time focusing on the generation of electronic music sounds with SP (Jáuregui, Dongo, & Couture, 2019). They discuss the building of invariant features in both cartesian and spherical coordinates, resulting in a heavy feature vector of 40 dimensions that is input to a decision tree classifier. They are then able to recognize 6 gestures with a performance of 68%[44]. Finally, they propose a user evaluation of their prototype in a learning scenario, pointing out the importance of the sound and visual feedback.

Both series of work on SP recognition propose minimal prototypes that can classify a little number of signs from a small database of training examples with important limitations in their work in the performance that they achieve and the assumption they make on what type of signs and be recognized.

4. Workflow and challenges

d) Initial planning

My initial master thesis calendar was the following:

- [Week 0] (10/02/2020): Soundpainting workshop with Walter Thompson @ EPFL. Work on Soundpainting grammar.
- [Week 1+]: Gloves and kinect acquisition and first tests with InteractML, Wekinator, Max/MSP to build a simplistic synthesis module. Getting to have a working, simple classifier with 2 hand gestures and 2 different sound outputs. Ideally, having the kinect working with the gloves already (both inputs).
- [Week 3+]: Starting the machine learning training of soundpainting gestures. Performance assessment and improvement of the ML module. Kinect and glove combination.
- [Week 5+]: Starting the implementation of a simple soundpainting parser based on the classified signs. Playing with simple requests and tweaking the synthesis module to produce more interesting contents/sounds than simple notes, according to the possible requests. Building of a simple video feedback.
- [Week 12+]: At this point, ideally, we should already be able to recognize simple requests and have an artistically interesting tool that can take sound inputs in different ways and switch instruments. Starting overall performance assessment, preparation of a small demonstrative performance. Starting the implemention of complex soundpainting rules.

[44] They use a penalized F-measure as main indicator of performance.

- [Week 15+]: Preparing final report & documentation. Optimization of overall performance and result quality. (Optional) Building a more advanced video feedback.
- [Week 17+]: End of the master thesis at EPFL. Starting the Cargo Bike Band 1-month trip through CH and FR, using the new tool! Augmenting the sound synthesis possibilities for band/live applications and continuous optimization of the setup during the tour.

e) The actual workflow

In practice, the workflow had been very different from my original plans:

- At the end of the SP workshop in February, I could already demonstrate to Walter Thompson the classification of two gestures (whole group, rest of the group) in a few minutes using Google Teachable Machine (https://teachablemachine.withgoogle.com) that functions with PoseNet. PoseNet already looked like the best candidate for a lightweight recognition pipeline on my laptop, which I could bring to the stage. Thompson would perform the signs a few times in front of the web camera, in very poor light conditions and Teachable Machine would still recognize accurately the signs, with some delay.

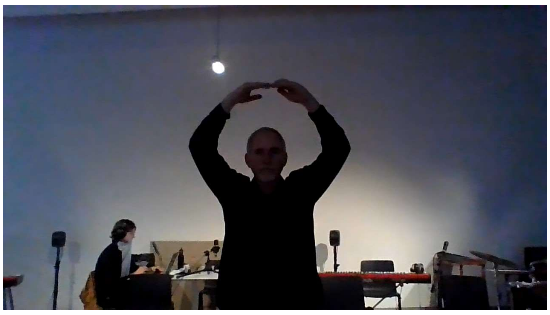


Figure 13 Example of a training image of "whole group" for Google Teachable Machine, taken during the first tests during the Soundpainting workshop at EPFL, 2020

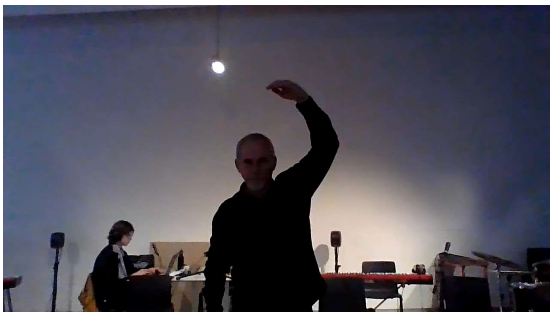


Figure 14 Example of a training image of "rest of the group" for Google Teachable Machine

- After two weeks of prototyping in Max/MSP and discussing with the InteractML team, I decided to drop Unity definitively and focus on a Max/MSP app.
- After three weeks, I could release my first demo video with PoseNet: https://www.youtube.com/watch?v=cFrR3W4-Tf4
- During the following weeks, I worked on:
  - the integration between Wekinator and Max to create an automated training pipeline and be able to train the models with a partner in front of the computer
  - the integration of an automata into Max and a first modelisation of the grammar of SP
  - the testing and integration of Kinect

As a result, I could demo the performance of Wekinator (versus the mubu.hhmm) and the first automaton draft in this video: https://www.youtube.com/watch?v=jW6bo6XkhFo

- The first of April, one week after the previous demo, I connected my Max app to Ableton for the first time and was able to control it with custom gestures: https://www.youtube.com/watch?v=OmPFMT9mgOs
  What I defined as a goal for week 12 in my original plan was already achieved then.
- At this point (weeks 8-9), I still did not use gloves. After some quick tests with Unity, I realized that they would require particular integration with Max from the C++ SDK before I could use them. Lorenzo Cantelli from the EM+ lab proposed to help me early April, allowing me to focus elsewhere.
- The month of April was the less productive one in terms of features:
  - I decided to go beyond Ableton's functionalities and rather find a tool that could simulate an orchestra beyond the launching of clips with a unique tempo, modification of the volume and some spatial parameters that come with Live. Instead, I focused on identifying a tool that can handle different tempos simultaneously.
  - I tried without success to compile the Max patcher into an executable app.
- Early May, I had identified the ideal Max package to simulate the orchestra in the Bach Project, built a data management system in Max to save the trained data and improved parts of the automaton.
- In the following weeks (11-13), I started preparing the management of multiple inputs in my system, hoping for the gloves to connect soon. I built the routing matrix of the input manager and improved the userflow and interface for the alpha release.

- From the week 13 up to now, I started focusing on the writing of this report and its theoretical part.
- In the meantime, I discovered the existence of the port of Google HandPose to Tensorflow and Max. I decided to import it into my project and make the first tests with both hands and full body. The first of June, I released a video introducing HandPose and the routing features of my tool: https://www.youtube.com/watch?v=rKD5BMaHml8
- With HandPose as a backup system to the gloves and considering the difficulties of integrating them in Max, I chose to drop the use of the Hi5 gloves for the final presentation and rather focus on the many theoretical aspects that my project had allowed me to think about.

B. The big picture

In this part, we are going to see the big picture of my SP recognition program. After pointing out to its github repository, I will give an overview of its structure and then motivate the choice of Max/MSP as the main host of this tool.

1. Github & Readme

The recognition program is shared in open source and visible on github at this link.

Here is an extract from the readme:

Soundpainting recognition tool

- Are you a soundpainter?
- Or do you want to control your live artistic digital setup with your own signs and gestures?

This tool is made for you!

As part of my master thesis at EPFL (Switzerland) at the EM+ lab, I am building an app with Max/MSP that allows the user to control a virtual orchestra with soundpainting or user-created signs.

[Features] I don't know Soundpainting, so what can I do with this?

Soundpainting is a sign language designed and used by composers to compose in real-time with multi-disciplinary performers (musicians, actors, visual artists, dancers...). Although it was first used and created by Walter Thompson, many other soundpainters have created their own signs for specific performances, just like you can do with this tool! Here is actually the list of features of my tool:

- Create and train your own signs. For instance, you could create a sign for "launch my program.exe". The program has been built to recognize common Soundpainting signs but you can just build the ones you need!
- Record your signs, save them to files and build your own dictionary of signs.
- Recognize signs using your own hardware: you can use the built-in motion capture models (PoseNet, HandPose, Hi5, Kinect…) but also connect your own with Max in little time. Once you have defined the number of features of your input, its name and connected the data pipeline to Max (for instance with OSC), you don't need anymore routing or spaguetthi patching! You can start building signs from very simple inputs, such as mouse, keyboards or your favorite midi controller if you want!
- Connect with as many Wekinator or Mubu machine learning models as you need for recognizing different types of signs: poses, movements, position in space…
- Create complex requests with the grammar of Soundpainting, which is optimized for real-time performance; but also creating your own regular grammar: your own sign language.
- Play with the built-in virtual orchestra or Ableton OSC controller and create your own set of sounds, triggers. With signs and gestures, you don't have to use hardware anymore to control you favorite DAW or software: your body can communicate with them.

The project has started with the following references in mind :

- MiMu Gloves https://mimugloves.com/ (extending music instruments with gestual controls)
- GeKiPe http://philippespiesser.com/projet/gekipe-geste-kinect-percussion/ (creating/extending music instruments with mechanical, percussion-like movements)
- Soundpainting as a standard, world-wide language for artistic performance, communication and composition http://www.soundpainting.com/

Each of these is an example of the performativity and potential of gestures and signs for music creativity, composition and instrument expansion.

What is the link with Soundpainting?

Soundpainting is a sign language that is used commonly between human perfomers, to communicate between each other or with a composer during the performance. For instance, using Soundpainting, you can form a request such as: "Guitar 1, improvise, with, jazz, feel, slowly enter" or "Dancer 2, make a loop, in relation with, guitar 1, now".

Now we all know how cool computer-assisted music or tools can be and the potential they offer. My experience with Soundpainting is that synthesizers, mixing devices, effects... can be painful to manipulate in real-time performance. With this tool, you can manipulate them directly from Soundpainting signs (or your own!): you can program them to respond to commands that you will be able to send with your body... and this recognition tool.

As for now, only basic parts of the Soundpainting grammar are implemented in this tool. There are plenty of modes in Soundpainting that could be added later in the future, that will allow to create different request structures. But if you are a beginner in Soundpainting and want to explore what you can so with some basic signs and your own sounds, this is the right tool for you. Then if you want to code your own regular language and get deeper in the interfacing possibilities, that's also the right place to start.

Setup

Required and recommended hardware

- Required: webcam OR low-latency external camera OR kinect input
- Required: 64 bits computer (Tested on Windows 10)
- Recommended with webcam input: Dedicated GPU (Tested on Nvidia 1060 GTX)
- Recommended: 16+ Go RAM, i7 or i9 CPU (Tested on i7-8750H)

Standalone app

Standalone apps will be released during the summer. Before that, you need to check the setup procedure (very simple) in order to access the source patcher and try it.

Using the project in Max/MSP

For now, this is the only valid procedure. It has some additional requirements, including Max/MSP that is not free, but with which I am building this app.

1. Install Max/MSP (latest version) https://cycling74.com/
2. *For use with Kinect input* Install Processing https://processing.org/ and launch the "simpleKinect" scripts in the "Utilities" folder.
3. Download/clone this repository
4. Go to the "Main_patch" folder and load the lastest version of the patcher into Max/MSP (.maxpat)
5. *For use with the built-in HandPose & PoseNet - webcam - inputs (recommended)* Install the dependencies by clicking on the dedicated button in the patcher. Then,

make sure that the Maxhelper process (Max Helper.exe on windows) as well as the electron processes run on the dedicated GPU and not integrated GPU, by checking your OS or GPU settings (if you have a Nvidia GPU, check its control panel).

6. Install the required Max/MSP packages: MuBu, Bach project (and whatever Max/MSP is telling you that you are missing, because I used several handy packages for Max)
7. *For use with Kinect input* Download the drivers for your kinect and launch the processing scripts located in the "Utilities" folder (check out https://github.com/jpbellona/simpleKinect).
8. Download Wekinator (Wekinator.org) and launch as many models as you use in the patcher?
9. You can now use the tool! For instance, try to launch the PoseNet model with the Wekinator DTW model and train your first signs!
10. To build your own standalone, check out the procedure for Max/MSP (and Processing if you use the kinect scripts).

Communication with Ableton Live

You can use the tool to communicate with Ableton Live.

1. Install Ableton Live 10 (it may be compatible with Ableton live 9), for instance the free trial version.
2. Install a compatible version of LiveOSC. It is recommended to use the following: https://github.com/ideoforms/LiveOSC. The LiveOSC folder must be copied to the "MIDI Remote Scripts" folder of Ableton. On windows: "\ProgramData\Ableton\Live 10 Trial\Resources\MIDI Remote Scripts", on Mac it should appear under: "/Applications/Ableton*.app/Contents/App-Resources/" (unverified). Then in Ableton Live:

- open File>Preferences
- under Link/MIDI, set Control Surface 1 to "LiveOSC"

References

This project is based on the following tools:

- PoseNet Node For Max: https://github.com/tejaswigowda/posenet-node-max
- PoseNet for dummies https://github.com/billythemusical/n4m-posenet-for-dummies and original N4M posenet https://github.com/yuichkun/n4m-posenet
- N4M HandPose https://github.com/lysdexic-audio/n4m-handpose
- SimpleKinect https://github.com/jpbellona/simpleKinect
- Wekinator http://www.Wekinator.org/

- Viz.js https://github.com/mdaines/viz.js
- Javascript state machine https://github.com/jakesgordon/javascript-state-machine
- The bach project (bachproject.net)

The project is licensed under GPLv3, as required by the Bach Project. Other licenses are less restrictive.

2. General overview of the system

The identification of the required features that was presented in I.E.2 is a direct inspiration for building the recognition tool with several independent layers, just like each structural level of SP, from the creation of signs to the parsing of its grammar. An illustration of all its components and features is given in Figure 15.

I conceived each layer as a specific function that the user should easily be able to identify and interpret. Inside each layer are different processes and objects that the user interacts superficially with from the interface of the program. For transparency, I have put in white the parts that I built myself and in blue the parts that come integrally or partially from other works[45].

At the interface level, all layers are implemented inside Max/MSP. The interface has a very basic look and design. The user can see the whole patcher in the main window and is also able to access specific functionalities of each layer by using tabs.

At the processing level, Max/MSP itself has three different threads that it uses for processing the data passing through its compiled objects. For these threads, Max guarantees the synchronicity/ordering of events. However, Max also interprets node.js code that is processed in external threads asynchronously to Max internal threads (see II.E.1 and II.E.6 below for some consequences of this remark).

In the following parts of the report, I will no longer refer to “performers” in the implementation of my tool but rather to “devices” that can be virtual instruments, controllers, etc, instead.

[45] External parts that are “shipped in” with the program are distributed under GPLv3 license (or less restrictive).

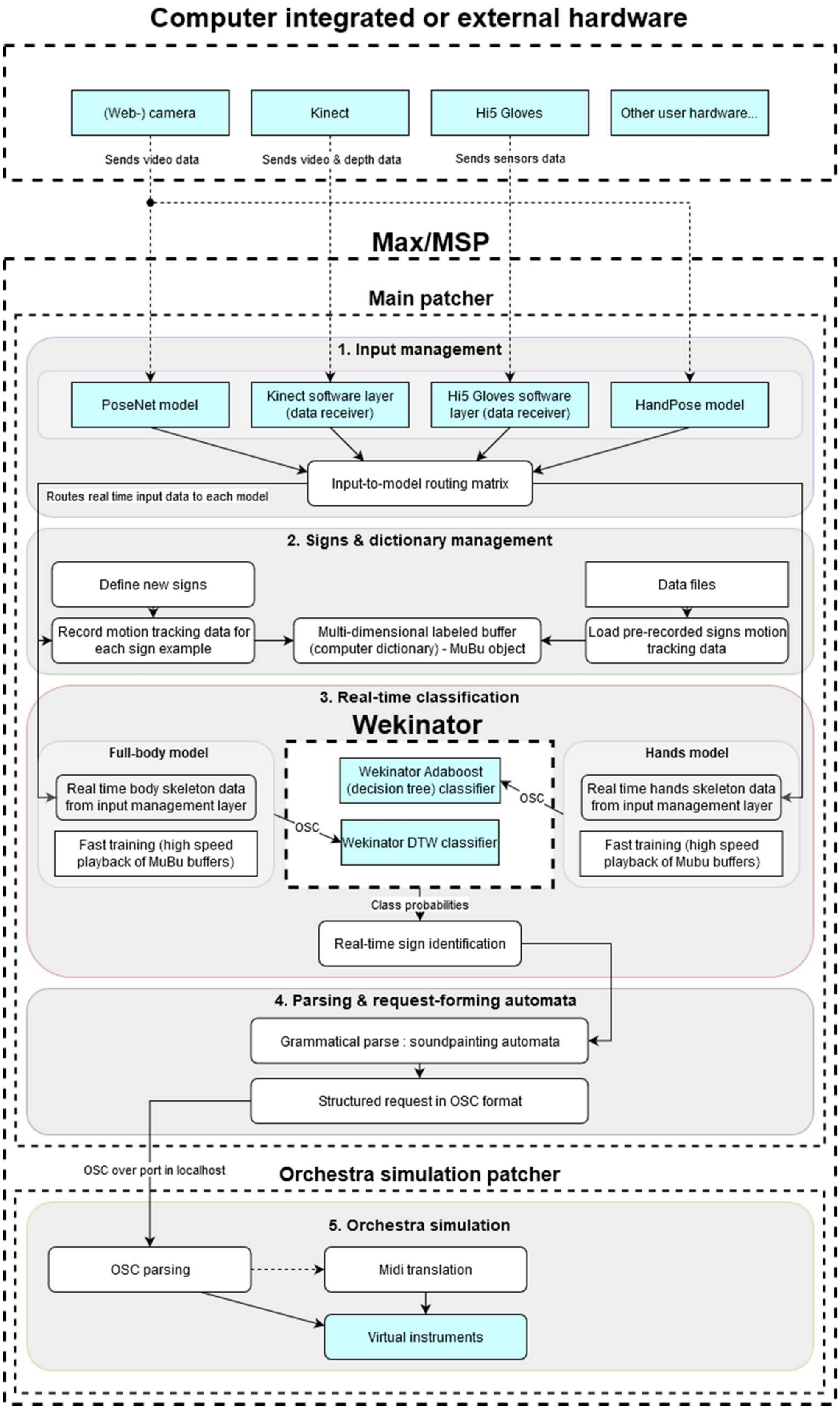

Computer integrated or external hardware
(Web-) camera
Kinect
Hi5 Gloves
Other user hardware...
Sends video data
Sends video & depth data
Sends sensors data
Max/MSP
Main patcher
1. Input management
PoseNet model
Kinect software layer (data receiver)
Hi5 Gloves software layer (data receiver)
HandPose model
Input-to-model routing matrix
Routes real time input data to each model
2. Signs & dictionary management
Define new signs
Data files
Record motion tracking data for each sign example
Multi-dimensional labeled buffer (computer dictionary) - MuBu object
Load pre-recorded signs motion tracking data
3. Real-time classification
Wekinator
Full-body model
Real time body skeleton data from input management layer
Fast training (high speed playback of MuBu buffers)
Wekinator Adaboost (decision tree) classifier
Wekinator DTW classifier
OSC
OSC
Hands model
Real time hands skeleton data from input management layer
Fast training (high speed playback of Mubu buffers)
Class probabilities
Real-time sign identification
4. Parsing & request-forming automata
Grammatical parse : soundpainting automata
Structured request in OSC format
OSC over port in localhost
Orchestra simulation patcher
5. Orchestra simulation
OSC parsing
Midi translation
Virtual instruments

Figure 15 Schematic overview of the recognition program in terms of different layers.

3. The choice of Max/MSP

Before starting a finer description of the mechanisms of the program, I would like to motivate the choice of Max/MSP as the main programming environment hosting the different layers and functionalities. There are several options for building such a tool, each with their own pros and cons. In my case, I have considered Unity3D and PureData as the main alternatives to Max/MSP:

- The advantage of Unity3D over Max/MSP(/Jitter) is that it is endorsed by a much larger community, it is free and has a higher potential in terms of graphics and interactive objects it can model.
- The advantage of PureData (PD) over Max/MSP is essentially that it is free and open source, while Max is a commercial software.

However, I think that Max/MSP is a better choice than those software for the following reasons:

- It is a high-level object-based programming language with already optimized pieces for music and real-time applications that are easy to use and assemble. It allows for scripting in JS and Node.js, which makes it a powerful host for a huge number of scripts and tools developed by the web community and its graphical interface allows newbies in coding to catch up easily with what is going on.
- Max/MSP comes from IRCAM and is used by a community of artists and musicians with interests very similar to my projects.
- Max/MSP has a nicer visual interface than PD and is maintained by a commercial company whereas PD has not been updated for years.
- PD connects badly to external pieces (node.js scripts, java…) that are critical for my project.
- The graphical programming interface of Max/MSP can be transformed into a user interface easily (for simple demos such as mine) … or very extensively, with complex Graphical User Interface (GUI) objects.
- Given my programming skills and the time constraints of the project, Max/MSP was a much faster approach than Unity3D.

Although being a commercial software, Max/MSP patchers can be compiled to a standalone program for Windows and Mac OS, allowing to share the program for free.

C. Description of each layer

1. Part 1: Motion tracking inputs

As humans, we are equipped with cognition and recognition systems that allow us to discern a wide variety of objects such as bodies in space and to build features to identify (classify) movements and gestural signs from those bodies. Computers, however, are not natively equipped with such systems, so that it is necessary to build them in this project, according to the goals and objectives defined previously.

The role of the motion tracking layer is to compute a set of motion features from the movement of the user. There exist several motion tracking systems with different technologies that can be used to model the human body from the position of characteristic points. These points can then be transformed into features of a classifier to identify gestural signs. In SP, there are some body parts such as the hands that are much more frequently to sign than others, therefore they require more precise tracking than the latter to classify amongst the signs. However, all motion tracking systems available have a finite range of operation, i.e. they can only track motion at a certain scale (just like the human cognition system).

We can observe two main scales at which signs are performed: full body and hands. Although there are certain costly technologies that would allow us to deal with both scales in one model, I propose and discuss two different technologies that are each adapted and efficient for each situation (body and hands).

a) Full body scale: “skeleton” tracking

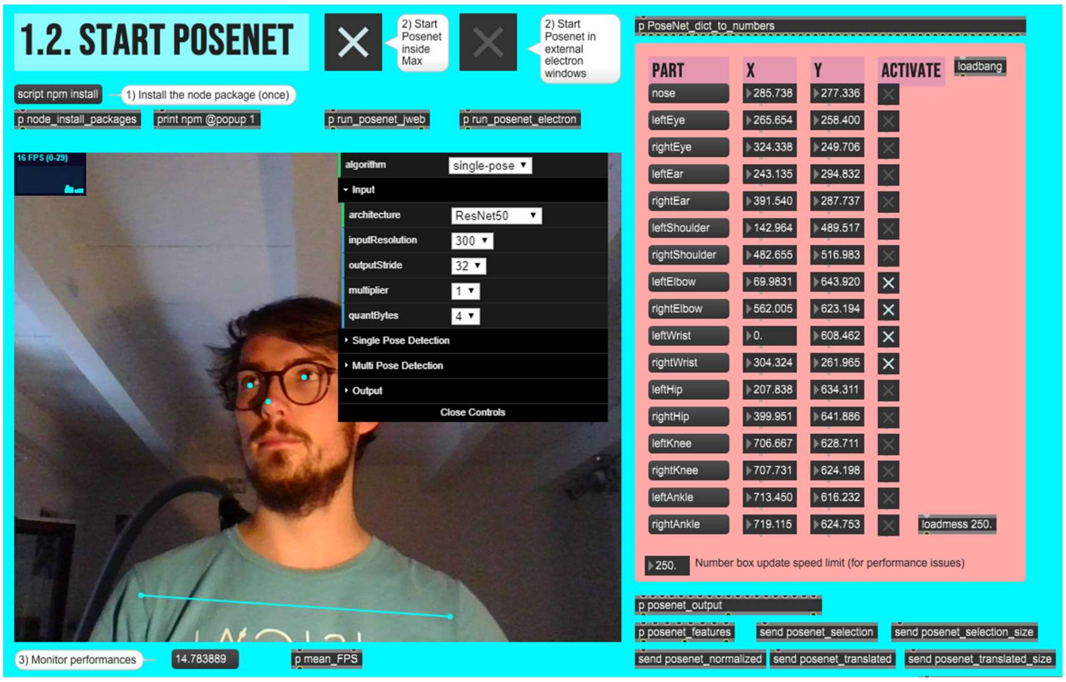
To construct our features for the identification of the signs, we must drop a lot of information from the input of the system, such as the webcam or motion tracking system we are using. For instance, information such as colors, certain body parts like the belly or torso, sound, etc, are not crucial for the identification of SP signs in the frame of this project. Typically, the “skeleton” representation of the position of each broad body part of the soundpainter (hands, shoulders, head, hips, knees, feet…) would allow the computer to recognize most basic signs at the full body scale. Because of the structure of the body (articulations, rigid body parts…), only several key points are needed to model its skeleton. Then, the features that would allow us to classify different signs must typically reach a precision of the order of magnitude below the distance between two keypoints. Assuming that most body parts are separated by a distance of the order of 10cm, we know *a priori* that our motion tracking model at this scale must reach a precision of the order of the centimeter.

In the following part, I will introduce and motivate the choice of PoseNet as the main motion tracking model at the full body scale.

(1) Introduction to PoseNet

PoseNet[46] is a computer-vision model that can be used to estimate the pose of a person in an image or video by estimating where key body joints are in 2D space. Its performances on modern CPUs and GPUs allow it to run in real-time using a webcam or alternative low-latency video input devices.

In Max/MSP, I could adapt and upgrade two demonstration projects showing how to port PoseNet into Max with Node.js to build a simple user interface (Figure 16) PoseNet separate Electron window) allowing the user to start PoseNet either in a separate window, within an Electron process (Figure 17 PoseNet separate Electron window or directly inside a Jweb object on the patcher. A demo of the process can be found in my second and third demo videos (march and April 2020) of the tool. I have observed a slightly lower performance with the Jweb object than the Electron process on my computer and the Electron process has the advantage of having its dedicated window, allowing the user to move it, resize it and most important, to keep it visible while the model is running, otherwise PoseNet performance drops critically. Both hosts (Jweb and Electron) are perfectly interchangeable in less than a minute (for loading the model).



algorithm: single-pose

Input

- architecture: ResNet50
- inputResolution: 300
- outputStride: 32
- multiplier: 1
- quantBytes: 4

14.783889

PART	X	Y	ACTIVATE
nose	285.738	277.336	☐
leftEye	265.654	258.400	☐
rightEye	324.338	249.706	☐
leftEar	243.135	294.832	☐
rightEar	391.540	287.737	☐
leftShoulder	142.964	489.517	☐
rightShoulder	482.655	516.983	☐
leftElbow	69.9831	643.920	☒
rightElbow	562.005	623.194	☒
leftWrist	0.	608.462	☒
rightWrist	304.324	261.965	☒
leftHip	207.838	634.311	☐
rightHip	399.951	641.886	☐
leftKnee	706.667	628.711	☐
rightKnee	707.731	624.198	☐
leftAnkle	713.450	616.232	☐
rightAnkle	719.115	624.753	☐

250. Number box update speed limit (for performance issues)

Figure 16 View of PoseNet GUI

[46] With TensorFlow: https://github.com/tensorflow/tfjs-models/tree/master/posenet.

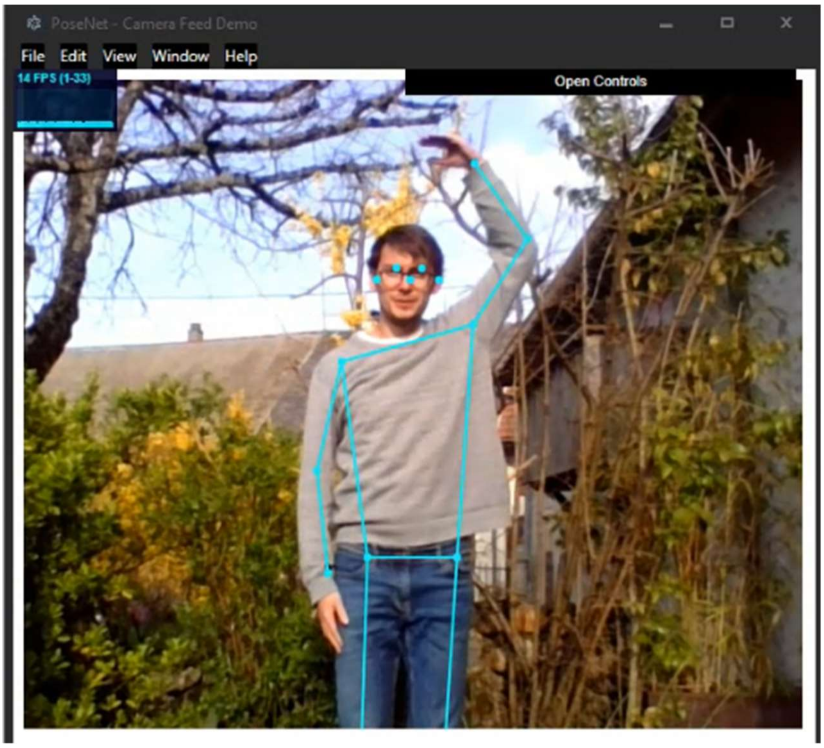


Figure 17 PoseNet separate Electron window

PoseNet allows the user to choose different models and internal parameters that will affect its performance:

- The architecture of the model (MobileNet or ResNet)
- The input resolution of the video
- The output stride of the model
- The depths of the convolution operations (for MobileNet only)
- The size of the model (only for ResNet, affecting loading time)

With these settings, the user can adapt the model to its hardware to get the best performance.

(2) PoseNet advantages

PoseNet has several advantages to its concurrent technologies:

- It works with webcams or any video input that can be recognized by the computer, so that:
  - For many laptops with integrated webcam, there is no need of external hardware

    - It can be used with very common and cheap hardware in case the computer does not have its own webcam, making the costs typically very low
- It provides a direct feedback of its accuracy to the user by overlapping the skeleton joints with the video, allowing users to change settings according to how good they see the model performing
- It is an open-source project, led by giants (Google…) and supported by a vast community
- It is still under development and will probably continue to be improved over the years, so it has a much greater potential than hardware-dependent solutions that are getting obsolete very fast
- It integrates with Max (and other systems) very easily as it can be run in a little node.js server

(3) Main shortcoming of PoseNet: depth

The only major shortcoming of PoseNet with respect to other motion tracking systems is that it only operates in 2D and does not model the depth of each body joint. As a workaround, I first built a simple calibration process that allowed me to compute the depth of the torso of the user as well as its angle to the camera. After some testing, I realized that it was useless in my case, that it did not allow any better classification and would only bring noise in the data.

(a) Depth in Soundpainting

There are specificities of SP that allow us to recognize its signs without any depth information from PoseNet. My observation is that depth (z axis) is often not the most informative axis for recognizing SP gestures and even signs like “play” which uses the z dimension extensively can be recognized only by the movement of the body in 2D, from the point of view of the camera in front of the soundpainter, because there is little ambiguity with other signs. In fact, depth would be helpful to differentiate two signs that look similar in a 2D space projection but is it to my experience never the case that signs are so similar that they require information about depth to be identified.

However, capturing depth is important for the two special signs in SP: “step in the box” and “step out the box”, which only takes place in the depth dimension. We have seen that the latter one - the first one is simply its complementary -, consisting in putting one foot in an imaginary box in front of the soundpainter, is used in many SP modes to signify “execute the request now”; whether the request has been defined previously (default mode) or is being signed while the soundpainter has stepped in the box. Not being able to recognize it is analogous to not looking at punctuation in a text: it may not be clear when the clauses or sentences end. We can then ask ourselves how to get past that specific issue.

(b) Recognition without depth: compromises and simplifications of SP grammar

Ideally, one would want to capture depth and abandon PoseNet for a better tracking method. In that case, seveval systems can be considered as alternatives:

- Kinect systems can capture depth but have many other shortcomings in terms of performance, user experience and portability
- Motion capture suits (for instance, IR marker-based suits) are usually the most accurate and performant devices but their costs and specificities make them unattractive for sharing the tool to the SP community
- OpenPose[47] is the main realistic alternative to PoseNet at this moment; it supports 3D triangulation from multiple view (like two cameras orthogonal to the soundpainter) but could not be ported to Max without much hassle[48]
- Some learnable triangulation methods are being developed recently[49] yet far from portable to my project

PoseNet appears like the best compromise for this particular use case, its goals and under the constraints of this project.

In theory, it would be sufficient to add a simple external hardware of software mechanism to know whether the soundpainter is in the box or not to remove the greatest part of the problem. One could for instance think of using a numeric carpet of a simple tracker of relative feet distance on the z axis to achieve this.

In practice, there are some light grammatical assumptions that we can make, which will allow us to overcome the problem of the recognition of the box. Let's remind us, for a moment, the category of signs called "go gestures" by Thompson. They do not appear as fundamental categories in my syntax analysis, but they are equivalent to what I call a "simultaneous" binome (timing, execution) (see I.C.2.b)(3)(e)). Even though the execution

[47] See https://github.com/CMU-Perceptual-Computing-Lab/openpose.

[48] The only realistic approach would have been to use the pytorch implementation of OpenPose (https://github.com/Hzzone/pytorch-openpose) and then try to run the python scripts in Max (Max does not interpret Python natively) with https://github.com/grrrr/py. It is very unlikely to work and could not be tested in the frame of my master thesis.

[49] The most convincing project is https://saic-violet.github.io/learnable-triangulation/.

sign may be used in other cases (for instance with a mode, with a content, with a modifier…), we can make a simplification of the grammar and assume for the first prototype of this tool that the execution is always related to timing signs. This is similar to Thompson's simplified description of the grammar in which the timing signs are called "go gestures", implying that they trigger the request. This simplification will allow us to simply launch the triggering of the signs each time a "timing" sign is given. However in future evolutions of the program, we will have to consider other means of recognizing depths if we want to implement other modes.

In conclusion, although more expensive or complex systems would allow for full 3D tracking of the body, PoseNet is suitable for building most features at the full body scale relevant to SP and is the most adapted technology in the frame of this project. From the observation of correlations between missing depth features and known signs/features, we are able to make a small simplification of the SP grammar by assuming that the requests must be executed immediately when "go gestures" are used, and that in every mode but the default one, the requests are immediate. Under these assumptions, PoseNet is sufficient to build all necessary features to recognize SP signs at the body scale.

(4) Building features

To build meaningful features for the classifiers, we need to take into account the following points:

- The set of features should reduce to the lowest number of dimensions, while still being meaningful, in order to avoid the so-called "curse of dimensionality" problem. It can for instance be solved with a principal component analysis which in general provides the best set of features to a given classification problem. However, in our case, we want the features to always remain interpretable to the user, so that we cannot afford such a transformation. Instead, we should only keep the most significant joints in PoseNet output such as the wrists or elbows positions, discarding less significant ones (for SP) such as the nose or hips. In case the user wants to build a custom set of signs that rely heavily on these body parts, he would however still need to be able to select different joints.
- The features must be invariant to transformations that are not meaningful and do not correspond to any sign. In particular, the features should be translation- & rotation-invariants and independent of the dimensions of the body of the soundpainter. In practice, this is achieved with PoseNet by taking the X & Y distance between each joint and the nose, which is considered a fixed point and then normalizing the X dimension with respect to the inter-shoulders distance and the Y dimension with respect to the distance between the middle of the hips and

the nose. The feature invariance will allow a sign to be recognized independently of the soundpainter's:

  - location and orientation (as long as the soundpainter faces the camera with a relatively small angle and does not get outside its field of view)
  - body dimensions (assuming that the general proportions of the human body remain constant)

With these constraints in mind and from my previous knowledge of SP signs, I could first order the body joints in PoseNet by their importance for SP signs recognition for basic signs in default mode:

- wrists
- elbows
- shoulders
- all other joints

In fact, I decided to only use wrists and elbows positions, which gave the best performance in my initial tests, reducing the feature space dimension to 8 instead of 34 (all 17 body joints X and Y values).[50]

b) Hands tracking

Similarly to the full body model, hands can be represented as a skeleton in which our features are built from the position of the hands in space. This time, a 3D model is necessary as several signs are ambiguous in 2D space and can only be classified by looking at whether the palm is facing the soundpainter or the opposite direction. It is for instance the case with the signs "two" and "volume".

(1) Hi5 Gloves

Just like for the full body model, I initially looked at several existing technologies to model 3D positions of the hands. At the time of my research (before early March 2020), there was no equivalent computer-vision model to PoseNet for the hands that I could

[50] I did not test the performance of this choice quantitatively, but only qualitatively by observing the change in performance that I could perceive with and without shoulders. It is obvious from the SP signs that I consider that aside shoulders, no other body joints could increase the performance of the recognition in real-time (curse of dimensionality problem).

integrate into Max.[51] Instead, a variety of dedicated hardware was already available on the market with three major technologies:

- Marker based gloves
- Flex sensors
- Inertial measurement units, typically using small magnetometers

The prices of these equipment is quite expensive on the market, from 1000 chf to 5000+ chf for most costly models. The cheapest models suffer from important shortcomings:

- IMUs are reported to drift when magnetic fields are present around the gloves, preventing a close use of computers, cellphones and other electronic devices nearby.
- Flex sensors are only one dimensional. In the cheapest gloves, many important dimensions of the fingers' movements such as the phalangeal joint angles or angles between two fingers are therefore not captured, whereas they are often used in SP and other sign languages.
- Some gloves are designed for specific software such as Unity and may not integrate with Max easily.

While most expensive gloves often provide solutions to these problems, I decided to start with the IMU based Hi5 gloves from Noitom, which are among the cheapest available on the market (1000 chf approx.) and provides a Unity and C++ SDK.

However, at the time of this report (early June), I have still been unable to test the gloves with Max. I made two mistakes during the conducting of the project:

- Before buying the gloves, I had not realized that their connection with the Vive trackers was not only a feature as they are marketed on the website and documentation but that the Vive trackers were also necessary for using the gloves directly with the provided Unity plugin. In fact, the connection with positional tracking devices was not explicitly documented as a requirement for the Unity Plugin in their documentation. We could therefore not use the gloves on the fly with the provided scene and needed additional scripting using the SDK.

[51] Several hand pose estimation models are presented here : https://xinghaochen.github.io/awesome-hand-pose-estimation/. The open pose model also has 2D hand pose recognition but in each case, the models could not easily be ported to Max.

- I had been late on testing the gloves and identifying that issue. Although it made sense to me to start by building all the core mechanisms of the tool and pipeline before integrating the gloves, it was a mistake not to test them earlier, so that I could have figured out this issue at early stages.

The EM+ lab offered me support to build a dedicated object in Max for receiving the gloves data. The first experiments with the C++ SDK were good but unfortunately, the difficult schedule of that semester did not yet allow us to go further.

(2) HandPose

Around mid-March, Tensorflow released the new HandPose model[52]. I first heard about it from within the Max community, when a first wrap of HandPose into an Electron server (just like PoseNet) was shared on github late May[53]. In the following days, I contributed to designing an interface within Max that made the output of the model accessible to the user (Figure 18).

While I was stuck with the Hi5 gloves, HandPose provided me an easy and light way to test the multi-input pipeline and the combination of the two scales or recognition inside Max. An introduction to HandPose inside my recognition tool is shown in my fourth demo video (June 2020)

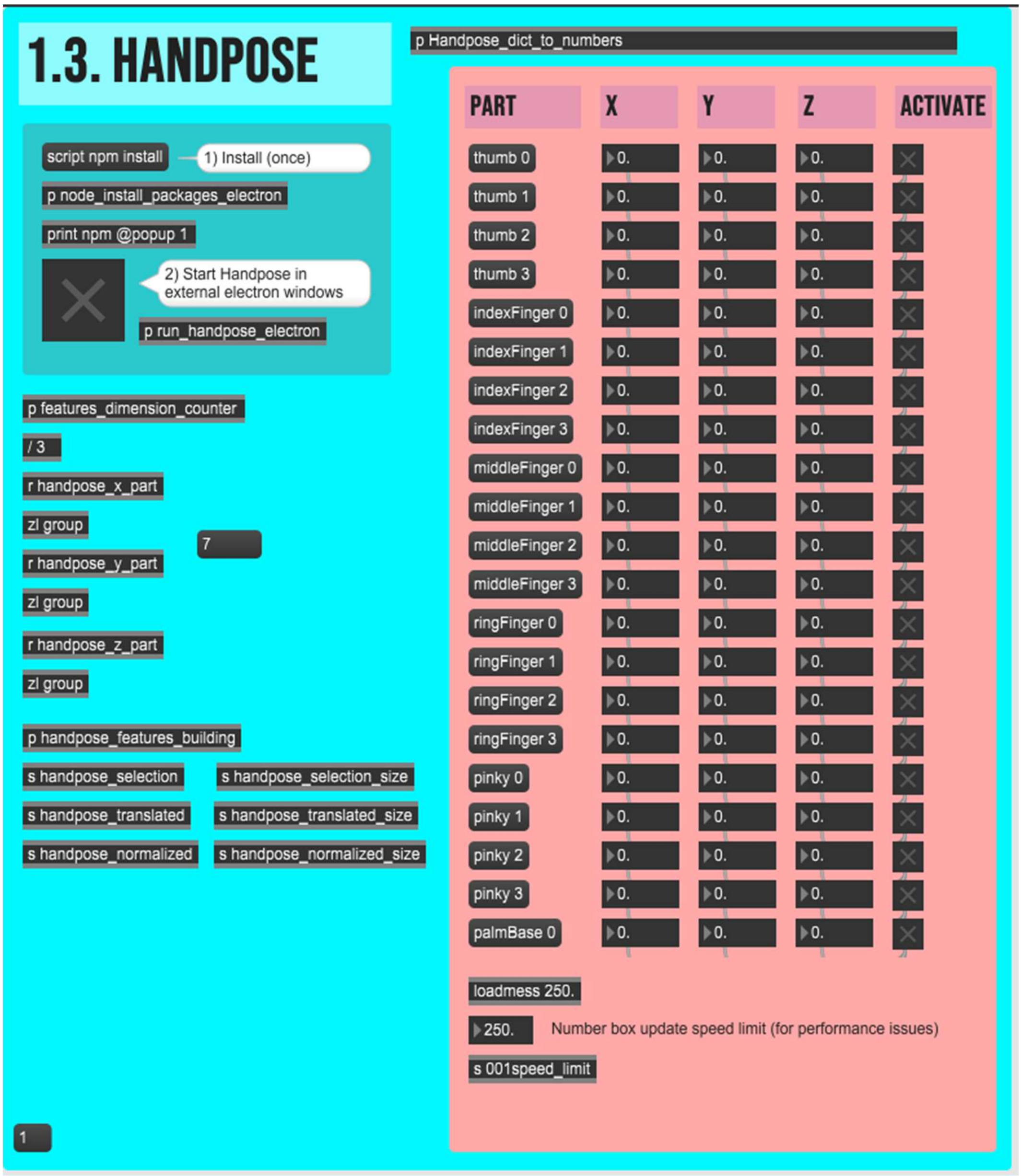
HandPose properties are very similar to PoseNet, hence it runs in a similar Electron node.js server, also has several performance settings and joints that can be selected by the user as features for the classification model.

Contrary to PoseNet, HandPose models the hand in 3D. However, it cannot yet recognize two hands at the same time, although there are good reasons to believe that this will be implemented soon.[54]

[52] See Tensorflow blog, March 09, 2020: https://blog.tensorflow.org/2020/03/face-and-hand-tracking-in-browser-with-mediapipe-and-tensorflowjs.html.

[53] See https://github.com/lysdexic-audio/n4m-handpose.

[54] See the MediaPipe repository: https://github.com/google/mediapipe/blob/master/mediapipe/docs/multi_hand_tracking_mobile_gpu.md.


1.3. HANDPOSE
script npm install
1) Install (once)
p node_install_packages_electron
print npm @popup 1
2) Start Handpose in external electron windows
p run_handpose_electron
p features_dimension_counter
/ 3
r handpose_x_part
zl group
7
r handpose_y_part
zl group
r handpose_z_part
zl group
p handpose_features_building
s handpose_selection
s handpose_selection_size
s handpose_translated
s handpose_translated_size
s handpose_normalized
s handpose_normalized_size
1
p Handpose_dict_to_numbers
loadmess 250.
250.
Number box update speed limit (for performance issues)
s 001speed_limit

PART	X	Y	Z	ACTIVATE
thumb 0	0.	0.	0.	☒
thumb 1	0.	0.	0.	☒
thumb 2	0.	0.	0.	☒
thumb 3	0.	0.	0.	☒
indexFinger 0	0.	0.	0.	☒
indexFinger 1	0.	0.	0.	☒
indexFinger 2	0.	0.	0.	☒
indexFinger 3	0.	0.	0.	☒
middleFinger 0	0.	0.	0.	☒
middleFinger 1	0.	0.	0.	☒
middleFinger 2	0.	0.	0.	☒
middleFinger 3	0.	0.	0.	☒
ringFinger 0	0.	0.	0.	☒
ringFinger 1	0.	0.	0.	☒
ringFinger 2	0.	0.	0.	☒
ringFinger 3	0.	0.	0.	☒
pinky 0	0.	0.	0.	☒
pinky 1	0.	0.	0.	☒
pinky 2	0.	0.	0.	☒
pinky 3	0.	0.	0.	☒
palmBase 0	0.	0.	0.	☒

Figure 18 My re-worked HandPose implementation max

In SP, signs that are made of hand poses can be performed in two ways, called the two-handed and one-handed versions. While HandPose only allows for using one hand at the same time, the soundpainter can therefore use only one-handed signs with the recognition tool. Another possibility would be to the split the video input from the camera for the left and right side of the body (only one hand in each side) and running the model on both sides. This has not been tested but would theoretically work without problem if the user is able to do so.

c) Input manager

I initially created the 'input manager' to allow the user to select in the list of all possible inputs (PoseNet, Kinect, gloves…) those that he wanted to use and that the program should listen to. This way, each part of the system would know how many features to expect and how to route the data flow correctly.

I started working with both PoseNet in the early stages of the project, I had thought that a single DTW model would be able to classify all signs from a sign input combining all the selected features from all inputs. My idea with the input manager back then was simply to concatenate all feature vectors from the different inputs into a single one, whose dimension would be equal to the sum of dimensions of all feature vectors.

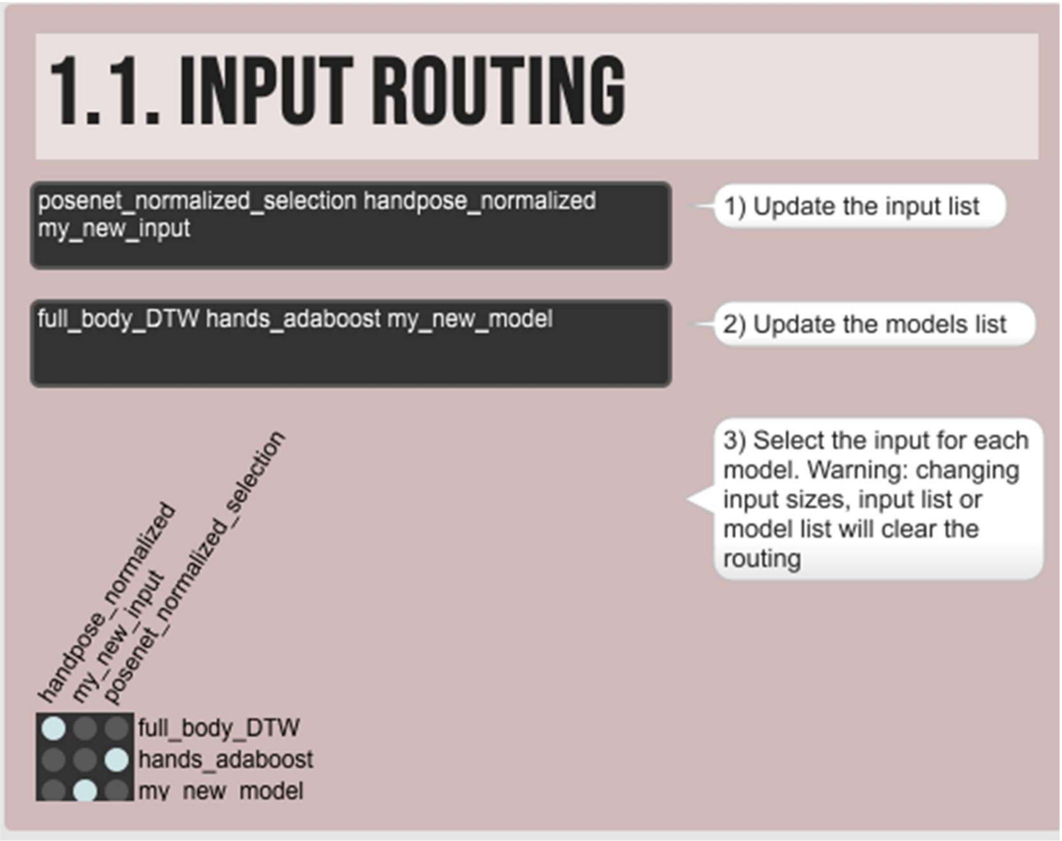
It is only once I started combining PoseNet and HandPose in the latest stages that I rethought completely the input to model scheme, in order to feed each model with a single input from a routing matrix. I opted for a modular approach with a design that allows the user to add its own inputs and models in from the Max patcher. Although there is no built-in routing matrix or system that would achieve this directly in Max, I had been able to dynamically create a matrix based on user input and route the data according to his selection. Therefore, I make no assumption on a particular model, feature vector dimensions or number of inputs and models in my program.

The present input manager allows the user to add his own input to the system automatically (Figure 19), without manually creating additional routings all over the program, by following a few conventions only:

- each input must send its data through a "send <input_name>" object
- each input must also forward its "size" (number of dimensions of the input) before the recording is launched[55] through a "send <input_name>_size" object

Then, on the input manager panel, the user can select what input should be used by each model. Because inputs have different data rates and dimensions, it is in general not possible to route more than one input to a single model. If two inputs are compatible (typically with similar data rates) the user simply can create a new input and merge the two original ones – in his own way - in a single one.

[55] The information flow at the opening of the patcher follows a specific order. If implemented at startup, the triggering of the input size should be launched with a « loadpercent 91 » object to guarantee that it is caught by the input manager. Otherwise, the number can be sent any time before the recording is launched.



1.1. INPUT ROUTING

posenet_normalized_selection handpose_normalized my_new_input

1) Update the input list

full_body_DTW hands_adaboost my_new_model

2) Update the models list

3) Select the input for each model. Warning: changing input sizes, input list or model list will clear the routing

Figure 19 Input to model routing interface. The routing matrix is automatically updated from the list of inputs and models that the user defines, allowing the user to add his own inputs or models in little time.

2. Part 2: Signs & dictionary management

Once the inputs and models are defined in the program, the user can start recording signs and building its sign dictionaries for each model.

d) Sign creation

We have seen that creating new signs is one of the core mechanisms of SP. It is also one important feature of my program in which the user can either create his own signs or use pre-recorded signs, for instance that would have been created or recorded by other users.

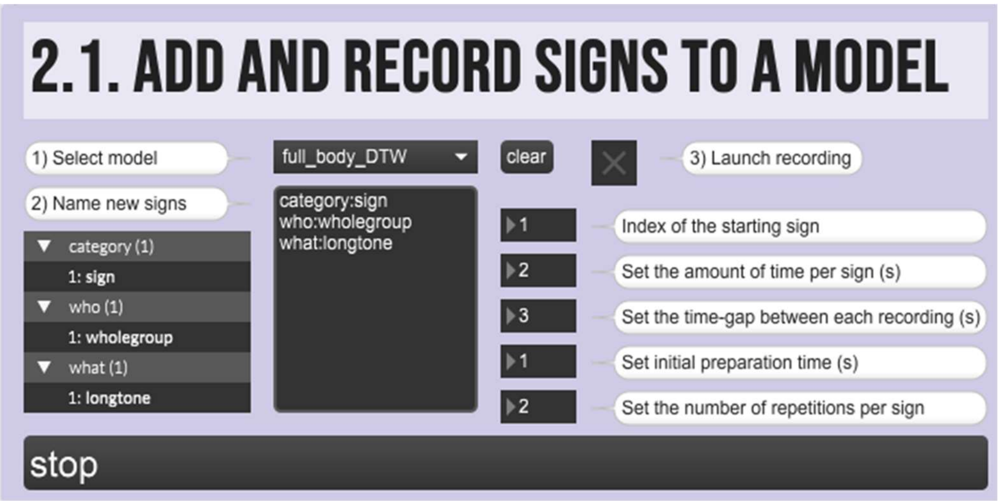
In my program, the user can define a sign by two properties:

- Its name

- Its category, in analogy with the syntactic model of SP: identifier, content, modifier, timing, off[56], neutral [57]

These properties are sufficient to allow the program to parse the sign, i.e. to construct a meaningful request from the temporal flow of signs.

The procedure for defining a new sign is shown Figure 20.



2.1. ADD AND RECORD SIGNS TO A MODEL

1) Select model: full_body_DTW

2) Name new signs

```
category:sign
who:wholegroup
what:longtone
```

▼ category (1)
1: sign
▼ who (1)
1: wholegroup
▼ what (1)
1: longtone

1 Index of the starting sign
2 Set the amount of time per sign (s)
3 Set the time-gap between each recording (s)
1 Set initial preparation time (s)
2 Set the number of repetitions per sign

Figure 20 Add and record signs to a model GUI.

In order for the sign to effectively be identified, two steps are required after the sign has been defined:

- Record training examples
- Program the virtual instrument itself to interpret the sign

While the recording of training examples is an automated process that simply involves pushing one button, the programming of the virtual instrument or device that the sign should control is outside the scope of the program. For demo purposes, I have implemented myself the interpretation of some signs by Ableton Live and a simulation of an orchestra with the Bach Project, but the connection with other devices must be implemented later by the user himself.

[56] Off is a mode, but I make here an additional simplification by considering it with a specific transition class in the automaton.

[57] Other categories such as MODE are not implemented yet, given that the program only focuses on the default mode of SP.

e) Sign recording

The user can choose to either define one sign at the time and record one or several training examples for it, then saving the training data and adding another sign… or directly define a list of signs and recording all of them in the same session. The recording session has the following form:

1. Initial preparation time of I seconds
2. Each sign is recorded N times, in the following loop:
   - The recording is launched for R seconds
   - Break (preparation time for next recording) of P seconds

The user can change the values of *0<I*, *0<N*, *0<R<5*[58] and 0<*P* according to his needs.

Each recording takes place in a different buffer of the Multiple Buffer (MuBu) objects (one Mubu object per model[59]) and each active input data is saved into a different track. The user can navigate the recorded data in each buffer and track using the Multiple Buffer Interface (Imubu object, Figure 21).

[58] 5 seconds is the maximum sequence length that I allowed the recording buffer to store. Internally, it is a constraint from the MuBu object that stores the buffers and require a « maximum capacity » for each track, possibly for memory allocation issues. However, it is unrealistic that a sign does take more than 2 seconds to be executed.

[59] In further releases of the tool, it will be possible to only use one Mubu object for all models, but I could not reach this level of genericity and complexity inside the frame of my project.

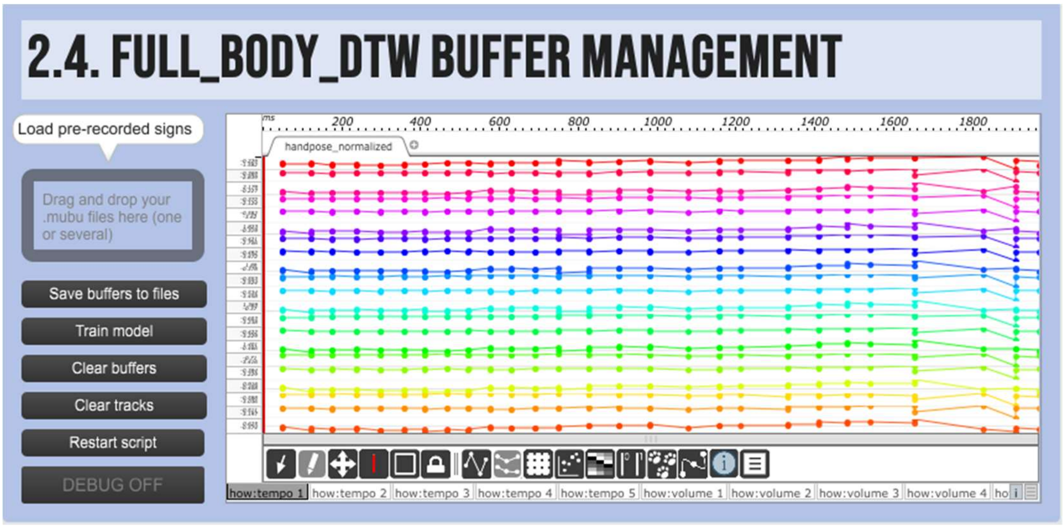


Figure 21 Example of the buffer management system for the "full_body_DTW" model. The user is able to control the mubu object through the dedicated buttons on the left. Each recorded example appears as a individual buffer (see bottom of imubu object), the motion tracking data is represented here by the colored lines and dots, and the corresponding input is the name of the tab (here "handpose_normalized")

Figure 8 Imubu controls and interface.

After the recording session, the data contained in a given track of a given buffer corresponds to a (N+1) x L matrix with N being the number of dimensions of the corresponding input and L being the number of steps in the recording sequence (sampling rate x duration of the sequence). The additional dimension corresponds to the time-tagging of the data. There are two main reasons for time-tagging the data:

- When several inputs are recorded at the same time, each has its own output rate; time-tagging the data in each track guarantees that during playback, the rate of each track is preserved
- Individual inputs can have varying output rates over time, for instance PoseNet and HandPose that run on GPU. Although we will see that the Dynamic Time Warping classification does "warp" the sequence in time and is therefore not sensitive to small variations of data rate, it is safe to assume that keeping the data timing in place always would better represent the original movements of the soundpainter

Wrapping up, inside the MuBu object, a sign is represented by labeled multi-dimensional buffers that contain the motion tracking data corresponding to each recorded example of the sign.

Once the data has been recorded, the user is able to save the recorded buffers to files in the ./data folder, by using the dedicated button "Save buffers to file".

f) Saving recordings to file: building a dictionary

What would be ideal is to store the data in the following fashion:

```
./data
   /track_name (corresponds to the input name)
         /sign_label + unique_id (.mubu or .txt)
```

The motivation for using this file and folder structure is that it best represents the data structure of the MuBu object itself and allows the user to clearly identify what the file corresponds to without looking at its metadata. The user can then mix data from recording sessions that are inhomogeneous, i.e. with a different number of recorded inputs, by loading all the files that correspond to the inputs he uses, even though they might come from very different sessions.

However, the MuBu object write and read mechanisms suffers from bugs[60] that should be fixed by the developers in the near future (as of May 2020) and I had to implement a workaround before it gets fixed, by saving each buffer with all its tracks in a single file:

```
./data
   /configuration_#track_name_1_#track_name_2…
         /buffer_name + unique_id (.mubu, .txt)
```

This way, the buffer names are saved correctly, but the user is no longer able to mix data from different sets of inputs.

g) Loading pre-recorded signs

Loading buffer data from files is much simpler and can be achieved in 2.3. by a simple drag and drop of one or several data files in the dedicated zone (see Figure 21).

h) Summary

Wrapping up, the user flow of the sign & dictionary management layer is the following:

- If the user wants to record new signs, i.e. either record examples of a sign that was not recorded and saved previously or record more examples of a sign that was already saved into files, he must first define which signs he wants to record and then launch the recording session.

[60] See: https://discussion.forum.ircam.fr/t/mubu-write-to-file-bugs-suggested-improvement/21714/2.

Once the signs are recorded in the buffer, he can save them to files by hitting the corresponding button if he is satisfied by the recordings.
If the user adds new signs again without saving the buffers first, the data that was contained in the MuBu object is lost.

- Once the recordings of new signs are finished, the user should load into the MuBu object the data files of all signs that he wants to recognize and classify, for all inputs that he would be using. This is achieved by dragging and dropping the corresponding files from the data folder into the dedicated zone.

3. Part 3: Real-time classifiers, regression or Dynamic Time Warping models

Now that the user has been able to connect his motion tracking inputs and record a few signs, we will see how the system is able to “recognize” the signs in real time with one or several machine learning models.

In the context of this master project and to offer the ability of creating new signs to the user, we must work with lightweight, interpretable models that can be trained fast and identify the signs that are performed in real time. In our case, the identification is a simple classification process, in which we ask the classifier to predict the “class” of the motion sequence performed among a set of classes that have been previously learned by the model - the SP signs.

We have seen that in SP, there are very different types of signs (movements, poses – at several scales). At the beginning of my project, I thought about recognizing all signs with a single Dynamic Time Warping model. Indeed, two light-weight models are generally presented in the literature to classify time-sequences: Dynamic Time Warping (DTW) and Hidden Markov Models (HMM). In general, DTW is observed to be faster and more accurate than HMMs (Raheja, 2015, Carmona J.M., 2012). Some works also propose combinations of HMM and DTW or modified DTW algorithms for gesture recognition (Hiyadi, 2016 and Choi, 2017).

My initial design was to sum up all the features that I would use (full body skeleton, hands skeleton…) into a single feature vector that I could feed to DTW.

The first obstacle to this initial design was that all inputs would not have similar data rates, so that the combination of inputs would either result in a data rate equal to the slowest input or a very high data rate, equal to the sum of each input data rate, but with common values between two consecutive feature vectors. This situation would not be ideal as it requires more processing and does not represent accurately the movement.

The second obstacle was that although DTW can recognize poses, it performs much slower than pose classifiers that are not time-dependent, such as SVM or decision trees. Ideally, one would want to construct one model per type of sign to be recognized.

I have chosen to implement two models for my final prototype: one for the full body with DTW and one for the hands with Adaboost decision trees.

i) Full body Wekinator Dynamic Time Warping

Unfortunately, at the time of the project, I could not find any real-time implementation of DTW in Max/MSP.[61] However, the external software Wekinator[62] offers a very efficient DTW implementation based on the FastDTW library (Salvador, 2004) with additional improvements for real-time performance and several internal parameters for its DTW model.

Max and Wekinator communicate in Open Sound Control (OSC). Although the user must launch Wekinator separately and perform basic operations on its GUI, most important parts of Wekinator can be controlled remotely via OSC, allowing Max to automatize certain operations, such its training process. The "user guide" for using Wekinator with the project is the following:

1) Start Wekinator.

[61]

- The MuBu library has a DTW object that can be directly used on the buffers but I could not find whether it could or how to make it work with real-time data.
- Frédéric Bettens from UMons presented in 2009 the num.dtw object for Max and PD, but it can no longer be found over the web as announced in its introductory paper *Real-time dtw-based gesture recognition external object for max/msp and puredata* in Proc. SMC '09, 2009 30-35.
- Another DTW object for Max has been built on the online-DTW library: *An Online Tempo Tracker for Automatic Accompaniment based on Audio-to-audio Alignment and Beat Tracking*, G. Burloiu. In Sound and Music Computing (SMC), 2016, but it is not designed to be used as a classifier.
- The RapidMax object (https://github.com/samparkewolfe/RapidMax) implements a part of the RapidLib on which is also based Wekinator, but has less functionalities and does not implement DTW yet.

[62] See http://www.Wekinator.org/.

2) Set the listening port to match Max settings and click "start listening"

3) Set the OSC input address to /wek/inputs (default)

4) Change the number of inputs (#inputs) to match the size of your input in this patcher, as defined in the first layer. For instance, with PoseNet, there are two features per joint (X and Y coordinates) so #inputs = #joints*2

5) Change the Wekinator output type to "All Dynamic Time Warping" with N gestures types, N equal or greater than the size of your dictionary of signs. It probably does not matter if you specify a greater amount of types, so you can also use any sufficiently large N if you do not know how many signs it should recognize; ultimately, Wekinator will simply never match the signs to those classes.

6) Set the output port to match Max/MSP settings and click next.

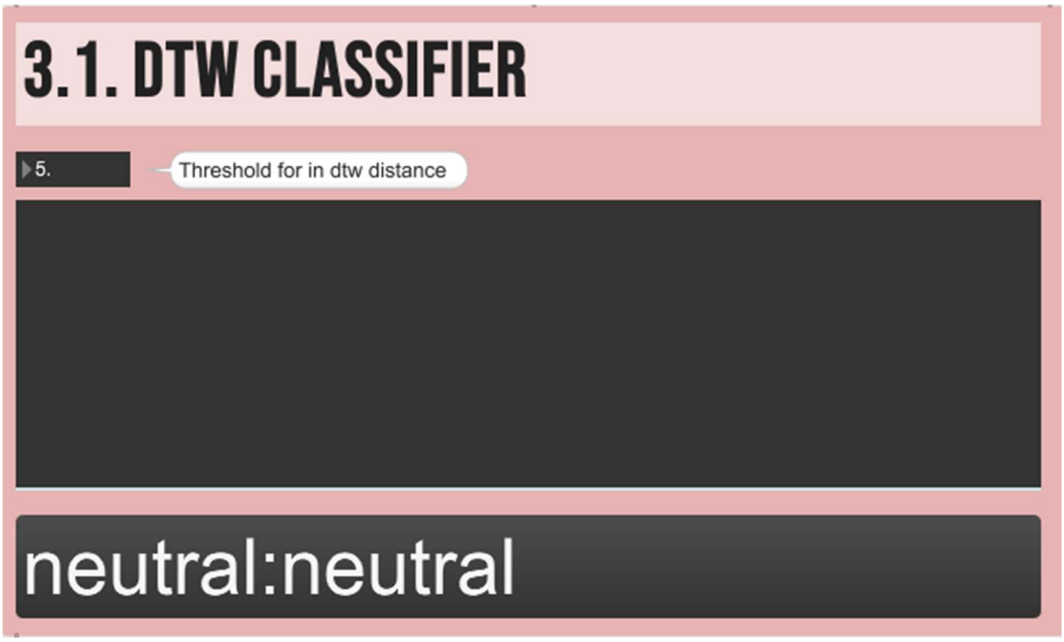
7) If any input is running, make sure that the "OSC In" indicator of Wekinator is green. If it is yellow instead, make sure you have some input running and try to open the view/OSC input status window and restart listening to the OSC. If it is red, check that the size of your input in Max matches the #input parameter of Wekinator.

8) You can now push the train button aside the Mubu object for the full body DTW model. The number of examples for each sign should show up in Wekinator.

9) Once the training is done after a few seconds, you can press the "run" button in Wekinator to start classifying your live input.

These operations may take 3 minutes at the first time use and less than 1 minute once the user would get acquainted with the process. Automatizing these steps would be very difficult from Max directly, as Wekinator builds its own file structures and I could not find a way to load a project in Wekinator from a command line directly. In the future, it is possible that the InteractML or RapidMax project will make the process fully automated.

Once the model is running, Max receives in real time the set of DTW distances from the real time sequence to each recorded sign sequence. By finding the minimal value in that set and comparing this value to a confidence threshold, we can identify when a sign is being performed in real time.



3.1. DTW CLASSIFIER

5.

Threshold for in dtw distance

neutral:neutral

Figure 22 DTW output GUI in Max. The middle black box are slides that show the confidence level of each sign in real time. Recognized signs are shown at the bottom.

We will address performance and accuracy aspects in a further section (II.D below).

j) Hands Adaboost for decision tree

To recognize hand poses that are used in signs like "tempo" or "volume", I chose to work with an Adaboost for decision tree classifier model in Wekinator. The configuration process in Wekinator is only slightly different and is fully detailed in my fourth demo video.

At the output of the model, Max receives the index of the most likely sign and can eventually threshold on the confidence of the classification to avoid false positives.

4. Part 4: Grammar parsing automaton

From the models introduced in the previous section, we can recognize individual signs, forming a sequence in time, just like words form a phrase in oral languages.

The next step is therefore to implement the grammar of SP with a parsing mechanism that would then allow us to create requests or commands to each device that acts as an individual performer in the system.

We have seen previously that grammars can be modeled with abstract machines called automata. In our case, there are context-sensitive components to the SP grammar that require us to use an automaton with stacks, i.e. arrays that contain information about previous states and transitions.

The most convenient way that I found to implement an automaton inside Max was using a node.js package called "Javascript state machine"[63]. By also using the Viz.js library[64], I was able to display the automaton inside a Jweb object in Max as a connected graph that represents all the states and transitions of the automaton. This is a very useful visualization for learning the grammar of SP and also for programmers to have a direct feedback on their grammatical implementation, for instance when adding new modes to the tool in the future (see Figure 23).

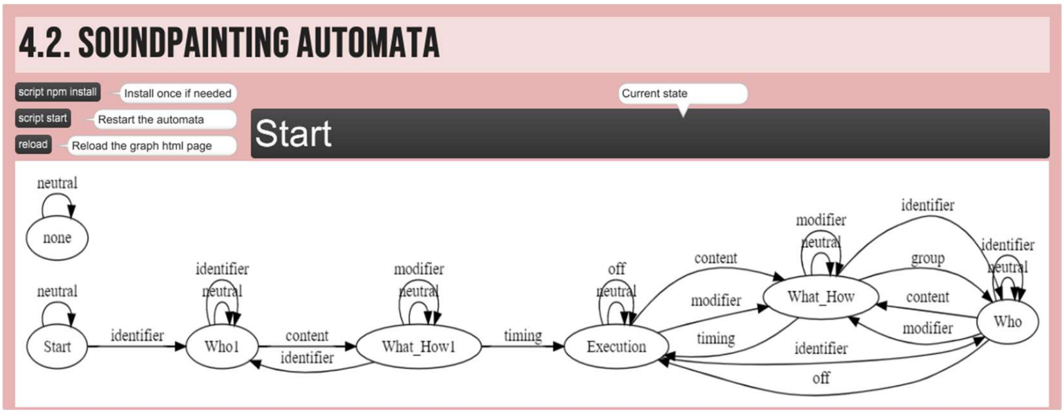


Figure 23 GUI of the Soundpainting automaton. The graph shows the different states of the automaton and the possible transitions between them.

Yet, the automaton only parses the default SP mode (by choice) without its prepositional elements. We have seen in section I.C.2.b) that prepositional elements introduce irregularities in the grammar which makes them complex signs to parse. Those features will be added in later releases of the program. The user can see the allowed transitions in the automaton graph (Figure 23).

k) Forming a request to devices from a sequence of signs

For constructing the request messages, I took inspiration from the OSC protocol: at the output of the automata, requests are sent to the devices in the recursive form

/device_name/content/parameter value
/device_name/content/parameter/parameter_of_the_parameter/…/… value1 value2 …

The way the request is formed inside the automaton is by collecting each sign during the state transitions and assembling them into several hierarchical objects:

- The request object (Figure 24) that stores each request in the following format:

[63] See https://github.com/jakesgordon/javascript-state-machine/.

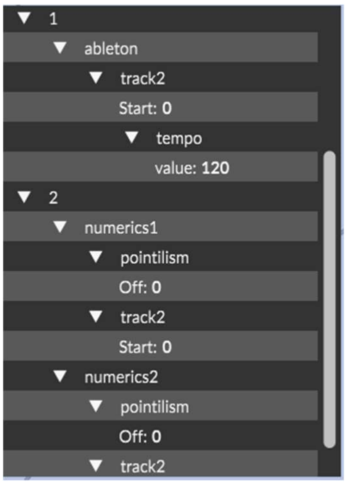
[64] See https://github.com/mdaines/viz.js.

    - At the first level, the index of the request
        - At the second level, the name of the devices
            - At the third level, the name of the contents
                - At the fourth level, the contents' parameters
                    - At deeper levels, the parameters of the parameters…

- The "distribution" object that stores what content each device is currently performing.
- The "reverse distribution" object that stores for each content, what device is playing it with what parameters.
- The identifier and content arrays that store temporarily during each clause the designated identifiers and contents.

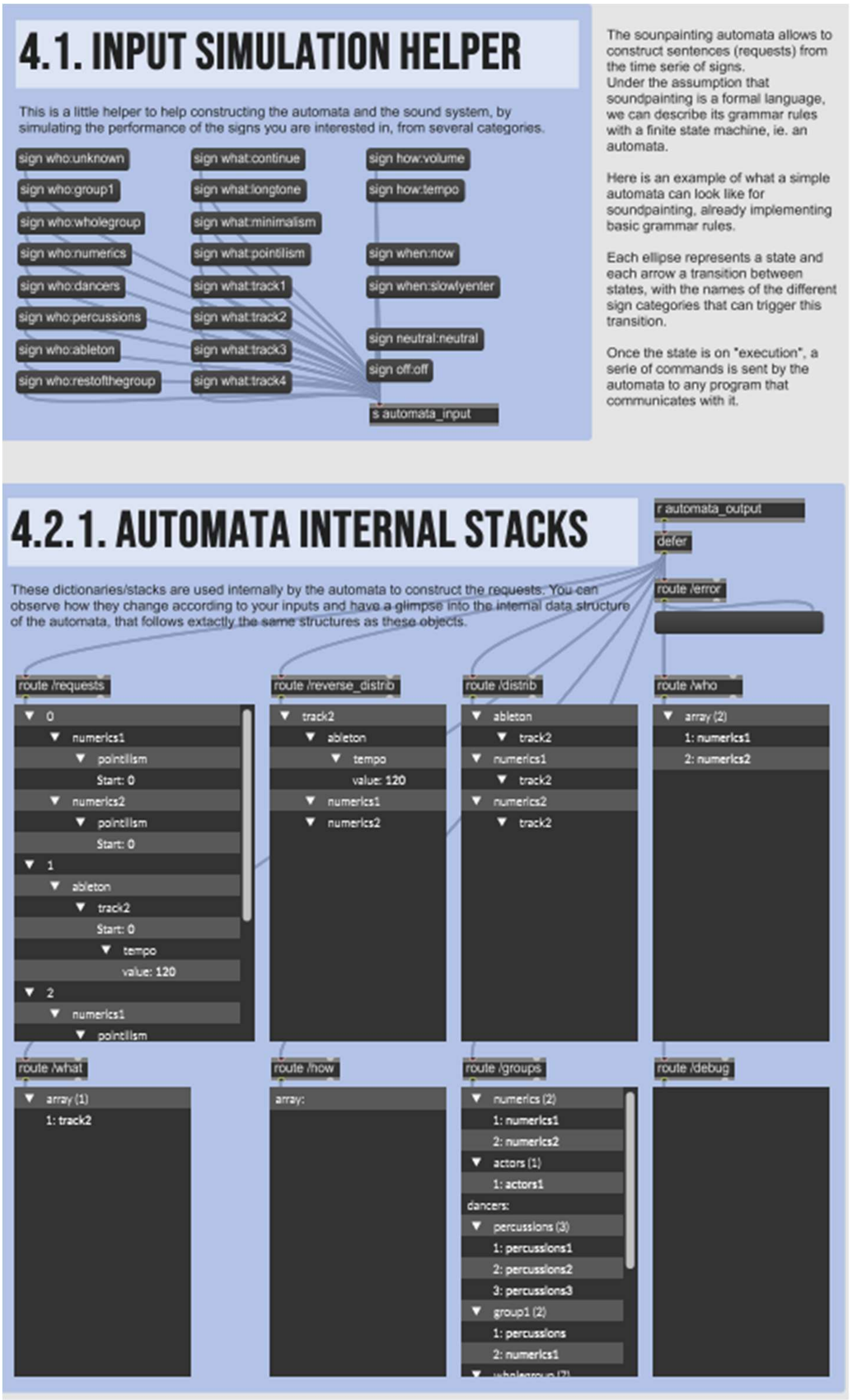
The last 3 objects are redundant with the request object (one could implement all the grammar only using the request object) but are more convenient to use and interpret by the programmer. Moreover, they provide a feedback to the user (for testing, debugging or understanding the internal mechanisms) on how the automaton is manipulating the signs and constructing the request internally.

Finally, the "group" object defines and stores all the groups of the performance and the "defaults" object stores default values for certain parameters. Both can be used to adapt the parsing to specific performance situation, in analogy with the default parameters and the conventional groups in SP.



```
▼ 1
    ▼ ableton
        ▼ track2
            Start: 0
            ▼ tempo
                value: 120
▼ 2
    ▼ numerics1
        ▼ pointilism
            Off: 0
        ▼ track2
            Start: 0
    ▼ numerics2
        ▼ pointilism
            Off: 0
        ▼ track2
```

Figure 24 The request object and its hierarchical structure, allowing for very generic representations of the parameters.


4.1. INPUT SIMULATION HELPER

This is a little helper to help constructing the automata and the sound system, by simulating the performance of the signs you are interested in, from several categories.

The sounpainting automata allows to construct sentences (requests) from the time serie of signs.
Under the assumption that soundpainting is a formal language, we can describe its grammar rules with a finite state machine, ie. an automata.

Here is an example of what a simple automata can look like for soundpainting, already implementing basic grammar rules.

Each ellipse represents a state and each arrow a transition between states, with the names of the different sign categories that can trigger this transition.

Once the state is on "execution", a serie of commands is sent by the automata to any program that communicates with it.

4.2.1. AUTOMATA INTERNAL STACKS

These dictionaries/stacks are used internally by the automata to construct the requests. You can observe how they change according to your inputs and have a glimpse into the internal data structure of the automata, that follows extactly the same structures as these objects.

Figure 25 Capture of the automaton debug and convenience panels representing each object (stack) internally used by the automaton

The details of the internal mechanisms of the automaton and the operations it makes on those objects are not described in this report because of the complexity of their translation in written English; they are available to the programmer by looking at the automata.js file that implements those mechanisms. However, there are specific elements of the code that can bring interesting points to a conceptual analysis of SP categories.

I) Discussion of some implementation choices and representation of Soundpainting concepts and elements

(1) “Play”, “slowly enter”: from the conceptual opposition between immediateness and delay to a continuous parametrization of time

To define “when” the events must happen in SP, it is very frequent to either use “play” for immediate requests and “slowly enter” for anything that needs to be delayed (and eventually precise the delay time using “within X seconds”). This way, there is a clear opposition between immediateness and delay, whereas for a computer, there are no such categories but rather a continuous parametrization of time, such that it is possible to synchronize events very precisely and define timing in ways much more complex than the immediate versus delayed dichotomy that is relevant to performers.

In the automaton, I implement the timing gestures (or go gestures) not as either immediate execution or delayed by a few seconds, but as a “start” command followed by a positive float number that represents the delay that is expected for the execution of the request. The play sign corresponds to a default timing of 0, while other signs such as slowly enter could be represented by an aleatory number in the range [0-5] seconds. Additionally, other signs could be implemented to modify that range, specify synchronous events, etc.

I think that conceptually, using machines and devices in SP brings new ways of thinking and working with time in SP. Because I am not an expert on that side of SP, I am not sure whether there are already signs for working with time at the precision and parametrization of machines (absolute timing, relative timing, continuous parametrization of time at the very low or very large scale…) but I have never experienced such signs. Of course, precision or parametrization is often not wanted in SP where the interest lies in the liberty that a performer has in his propositions. However, with machines, fine descriptions of time are often relevant and allow for a wide variety of results (think about signal processing, effects…). I find the idea of requesting a computer to delay a sequence by a few samples only or reacting synchronously to events very interesting and powerful. I hope that the language could evolve from and with the use of such technologies.

(2) Default parameters

Similarly to the default timing properties for “play” or “slowly enter”, default parameters can also be implemented for all modifiers inside the “defaults” array. This is an attempt to represent the relativity of faders with respect to a “default” or medium value.

(3) Content and modifier sharing the same state in the automaton: the “sculpting state”

Originally, I had separated the state “What” and “How” in my automaton, before observing that after the first request, they shared the same transitions and could be merged into a single state that I called “What_How”. One could interpret this as the conceptual proximity between content and modifiers as “sculpting gestures” and think about renaming that state the “sculpting state”. We have seen production rules that allow to form a modifier from a content followed by the postposition “group”. We can also observe a conceptual proximity between modifiers and contents as we push the transformations of contents to the extent that they can no longer be recognized as the original contents. A simplistic example is that lowering the volume of any sound (long tone, minimalism…) might result in a silence, which is conceptually different from those contents.

On one side, SP proposes to conceive a content as a binome (prototype + parameters), around the prototype that has “default parameters”, such that signing a content leaves implicit parameters. On the other side, one could push the limits of the “modifier” very far, such that at the end, the content has totally changed. It shows that the frontiers or the concept of content and modifiers are very porous, just like we have seen in the theory of concepts previously. The fact that the content and modifier state coincide is to me a direct consequence from the porosity of these two concepts that can regroup under the single idea of “sculpting”, whether it is at a fine level (subtle changes in volume that are close for the human perception) or broader level (change from a long tone to a pointillism, that are very different conceptually).

(4) The challenge of adpositional elements

We have seen in the SP syntax that using prepositional elements can change the function of the signs. For instance, in the sentence “whole group, movement with pointillism, slowly enter”, pointillism is a content but the phrase “with pointillism” performs the function “how”. Let us now consider the following example:

- First request: “percussion 1, actor 1, dancer 1, minimalism, play”
- Second request: “numerics, long tone, play”
- Third request: “minimalism…”

At this point, the automaton cannot determine whether minimalism is part of the broad identifier “minimalism group” or if it is a content for the identifier “numerics”. To parse the syntactic role of “minimalism”, we therefore must know more about its context, here what precedes it. This prevents us from using a simple finite state machine to parse those signs. Instead, stacks that represent the history of the automaton and its contexts can be used to parse such examples. In my prototype, I have not however implemented the mechanisms to parse adpositions yet.

(5) Omissions and restrictive rules about content or identifier repetition

I have been able to implement the omission mechanisms of SP, such that it is possible to use modifiers or contents without repeating the identifier (and the content in the case of modifiers). In practice, this is implemented by looking at the “distribution” and “reverse distribution” arrays that embed contextual information.

Moreover, I have implemented the restriction rules about repetition of identifiers in the same sentence, such that wrong sentences like “Percussion, LT, percussion, minimalism, play” are indicated as such (an error is provided to the user).

(6) Groups and dynamic groups

Finally, I would like to mention the implementation of groups inside the automaton. I separate them in two types: conventional and dynamic ones.

The conventional ones are groups such as “strings”, “brass”, “singers”, “visual artists”, etc, that are determined by the orchestra itself as part of the configuration. They are implemented inside the “group” array, in a hierarchical way, such that groups can contain other groups (but it should not contain circle-references, i.e. that group A contains group B that contains group A…). The moment that their signs are encountered, the groups are decomposed into their most basic elements, i.e. each device, such that they are not explicit references in the request itself.

The dynamic ones are groups such as “whole group” and “rest of group”, that are not defined in the group array *a priori* but are computed dynamically when they are encountered. “Whole group” is simply computed as the set of devices that are contained in the group array and those that were also mentioned previously, even if they are not encountered in the group array.

5. Part 5: Orchestra simulation

From the OSC commands created by the automaton, there is an unlimited panel of tools and ways to create an orchestra.

m) Digital Audio Workstations

DAWs are the most common programs in the music industry. They are mainly designed for recording, arranging, editing and mixing music sessions, but some of them also provide a range of tools for live performance and connectivity to midi or OSC-capable devices. What makes them the most interesting programs to control from Max is that they are extensively used by musicians and artists and most host virtual instruments (VSTs) natively. There are pros and cons for simulating a SP orchestra inside a DAW.

On one side, I would simply need to send midi or OSC messages to the DAW to launch pre-made clips and control simple elements like tempo, volume, etc. DAWs offer two high-level and ergonomic features that are not found into Max:

- Midi “piano roll” editors that allow for creating sequences by placing midi notes on a virtual keyboard, changing their on- and offsets, copying and pasting groups of notes, changing swing, quantization, volume and other parameters.
- A timeline that allow for organizing clips (sound or midi items) and arranging them in time.

For my program, I considered these features critical for the users to be able to customize the set of sounds, clips or virtual instruments that simulate the orchestra.

On the other side, DAWs do not have the flexibility that would be required for composing in real-time with SP gestures:

- Working at several tempos
- Looping things on the fly
- Working with continuous frequencies (in contrast with quantized pitches in the frequency domain) for glissandos and pitch manipulation
- Extension to generative methods that rely and real-time analysis of musical elements
- ...

For many of these elements, Max/MSP looks like a better choice than DAWs because of its packages and objects dedicated to these features.

n) Ableton Live

I originally started by building a connection with the performance software Ableton Live because of its widespread use in the community of live performing artists (DJs, live performers) and its useful features for live music such as working with preset scenes and launching synchronized clips based on a particular metric and tempo.

Just like many DAWs, Live can be hard to control externally and only offer a limited API to its internal mechanisms. Controlling Live externally typically requires scripting a server in python that must be installed manually. However, LiveOSC[65] has been shared by the community to help in this process by creating a server that listens to OSC map it to Live parameters.

In my third demo video, I showed how I could use custom gestures to launch and stop clips in Live, just like a simple DJ controller would. It required only a little effort to convert the conventions I use in my OSC commands to match the LiveOSC format. By synchronizing all the clips at the same tempo and metric, Live allowed me to create a fun demo with very simple controls that "sounds good" and can easily be modified and customized by the users.

However, the integration of SP commands could not be pushed very far into Live without more complex scripting by extending the LiveOSC API, or perhaps by using additional custom Max for Live scripts. For instance, in Live like in many DAWs, it is not possible to change dynamically the tempo of single elements, as they are built around a single timeline whose tempo is usually shared by all clips. This is a very useful and ergonomic design choice for many use cases but also a limit to one of the most basic elements of composition in SP. Extended research showed me that there are some workarounds to change the tempo of a scene from its name, but getting into those modifications in real time would require a very deep "hack" into Live. Given that Live is a proprietary software that would maybe not be the most popular choice among the community of performers and Soundpainting that I target with my tool, I chose not to dedicate it more time in the frame of this project;

o) Reaper

After my tests with Ableton, I decided to experiment with Reaper, which is a much more affordable DAW and often used for mixing, editing compositions, rather than live performance. However, Reaper does midi, OSC and allows for custom scripting in several programming languages for an extensive control of the program. However, there is no easy way to assign objects to different tempos dynamically in Reaper. I have considered two options:

- The first and most simple option would be to have one instance of Reaper for each instrument. This way, the tempo of the instrument could simply be controlled with

[65] The version that I am using is the following : https://github.com/ideoforms/LiveOSC.

the tempo of the timeline, which is easily accessed in OSC. However, this is a huge load on the CPU and not an optimal choice in terms of ergonomy.

- The second option would be to use a hierarchy of subprojects inside Reaper, each with their own tempo. The parent project would contain the whole orchestra and divide in sub projects that each represent a different virtual instrument. Each subproject would then again be divided into sub projects that each represent a different content. This way, the soundpainter could control the tempo of each content separately. However this option is painful to structure and modify in terms of ergonomy and settings.

After some testing with each option, I decided to abandon Reaper for other tools.

p) OSSIA Score

OSSIA (Open Software System for Interactive Applications) Score was for a long moment my major lead for implementing the orchestra simulation. It is an open source project for building interactive applications linking several independent elements that can communicate in Open Sound Control (OSC): Max/MSP, Unity, Motion capture systems, VJ systems, PureData, openFrameworks etc. Score is the main program of OSSIA and allows users to sequence events and processes in a kind of extended score. The events are represented generically (without explicit reference to whether they are notes, images, frequencies, midi commands…), allowing for the connection of very different types of data inside a non-linear timeline. Although many scoring notation systems such as DAWs make the assumption of a linear arrangement of events, Score has conditional triggers, independent loops and specific rates or tempo attributes to each process: OSSIA Score is a major breakout to traditional scoring systems that fits exactly our need for simulating the orchestra.

The idea would be to input the OSC from the automaton into Score to trigger the events in the score dynamically. The events could be sound files as well as midi clips, that would then be sent to virtual instruments, for instance into another Max patcher.

However, OSSIA Score is in development and has a very painful userflow/ergonomy to my appreciation, to the point that I decided to wait for later releases before spending too much time on achieving very low-level actions. During one month, I have been in contact with the developers to report some issues or suggestions on the interface and userflow. I believe that in the following release, it will have a great potential with my tool for controlling video devices, robots, music systems and much more with gestures.

q) The Bach Project

The Bach Project[66] is a package for "computer-aided composition in Max" which provides a family of open-source tools for music notation in Max: Bach, Cage and Dada.

Bach implements both classical music notation and proportional music notation, with support for accidentals of arbitrary resolution, rhythmic trees and grace notes, polymetric notation, MusicXML, MIDI files and much more. The graphic position of all notation elements can be queried so that reactive customized notation systems can be built.[67]

Whereas I had originally considered it a too low-level system for the simulation of the orchestra, I have finally realized that Bach provides in Max the set of tools that I was searching in DAWs and was the most adapted one to implement a very deep control of the orchestra with.

The orchestra simulation with Bach is implemented in a different patcher than the recognition tool and receives the commands from the automata with OSC. For the prototype, I have decided to build fixed contents into the bach.roll and bach.score objects that allow the user to place notes into a score, just like any score editing program. These objects are then connected to virtual instruments through midi. Unlike Ableton, Reaper and DAWs, the bach objects can be extensively tweaked from Max and each has its own timeline, tonality or micro-tonality, tempo, volume, dynamics, etc. In fact, almost everything that can appear on a traditional scoring environment is integrated in the Bach.roll and Bach.score objects. The main difference between the two is that the score object has a metric and displays the notes with rhythms, whereas the roll object is completely independent of any rhythmic system.

D. Performance aspects

1. GPU settings

On most computers, it is necessary to modify the default settings of the GPU to make sure that PoseNet and HandPose run on the GPU instead of the CPU graphics. When running the models in the electron windows, it is the electron processes that must be set to run on the GPU. When using PoseNet in Max Jweb object, it is the Max helper process that must be set to run on GPU. For that reason, I highly suggest using the electron process whenever possible, so that other computations in the Max helper process do not

[66] See https://www.bachproject.net/

[67]See https://www.bachproject.net/features/

occur on the GPU. By default, it seems like all node.js script are run within the Max helper process.

With a Nvidia 1060 gpu (Max-Q), 10 FPS for PoseNet is typically achieved with ResNet50 at quantbytes = 1 and input size = 350. With better GPUs, the PoseNet settings can be increased to keep the FPS below 15 while achieving the best accuracy.

2. PoseNet and HandPose settings and limitations

a) Settings

In the early stages of my tool, I was only able to use the most basic PoseNet model MobileNet V1. In April, I could upgrade it to the bigger and more accurate ResNet50 model, which made a huge improvement in its performance. I suggest that the user always set (on the GUI) the model to ResNet50 and 4 quantBytes, which corresponds to the highest model size.

On the side of HandPose, there are no settings yet (they are disabled for compatibility reasons), but similar configuration possibilities should be added soon.

b) Light environment

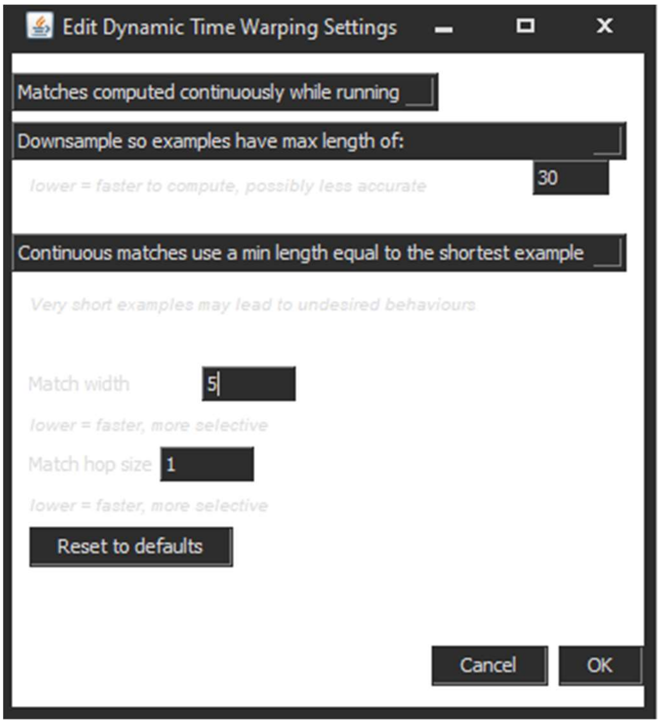
Both HandPose and PoseNet are very sensible to light conditions and contrast. Although it is difficult to describe the perfect environment in those terms, the user should pay attention to both camera settings (luminosity or ISO, contrast, saturation profiles, etc) and the position of the lights in his configuration to ensure that the models can work with best accuracy. A rather uniform background will probably result in good recognition when in high contrast with clothes and skin color.

c) Distance to the camera

Both models are limited by the size of the user's body or hand inside the field of view of the camera. Whereas PoseNet can recognize the full body at greater distances, HandPose is that the model only works as long as the hand is sufficiently close to the camera, typically within 2 meters from the camera. This greatly limits the ability of the soundpainter to use PoseNet and HandPose models on the same video source. While PoseNet has a greater accuracy when the full body is visible (not only arms), HandPose requires the user to typically get closer to the camera than the distance at which the whole body fits inside the camera's view. I suggest using a separate camera for each model, one that would be close to the soundpainter for HandPose and one further for PoseNet. I have however not been able to test this configuration yet. If it is not possible, then I suggest making sure that the arms, torso and head are well visible for PoseNet, while leaving hips, knees and feet under the field of view of the camera and making sure that hands are put as close as possible to the camera when signing with hand poses.

3. PoseNet ideal FPS

As mentioned in REF, PoseNet has different parameters that influence its performances. It is however not obvious where lies the best compromise between accuracy and the number of poses per seconds (FPS). Let's take a look at how Wekinator processes its input to identify the best settings for PoseNet.



Matches computed continuously while running

Downsample so examples have max length of: 30

lower = faster to compute, possibly less accurate

Continuous matches use a min length equal to the shortest example

Very short examples may lead to undesired behaviours

Match width 5

lower = faster, more selective

Match hop size 1

lower = faster, more selective

Figure 26 Wekinator DTW settings

We know that by default, Wekinator's DTW is downsampling the data rate to improve the DTW speed (Erreur ! Source du renvoi introuvable.), such that the best compromise in performance and accuracy is to keep the number of FPS just below what Wekinator can handle without downsampling for a sequence of 2 seconds[68]. With the DTW default settings (max sequence size = 10), the ideal number of FPS is 5[69]. On a fast

[68] 2 seconds is the default value of the recording sequence for each sign. The implementation details of Wekinator are discussed in the course *Machine learning for musicians and artists, Working with time* on Kadenze.com and visible on https://github.com/fiebrink1/Wekinator.

[69] We indeed have FPS*sequence_length = sequence size.

computer, it would be worth changing Wekinator's default max sequence length to 20-40 and run with around 10-20 FPS, which have proven to be more than enough for SP recognition or disable downsampling.

4. Wekinator Dynamic Time Warping

By setting Wekinator DTW to use a minimum sequence length of the shortest sample, it is a good practice to ensure that each training example recording is made with approximately the same range of input data rate. Indeed, using training examples that are very short (ie, very low number of points in the sequence) will lead to a great loss in accuracy from Wekinator. If that is it the case, I suggest modifying the minimum length for matches to around 15 samples.

To ensure a good quality of the recognition with DTW, I suggest recording more than 10 examples per sign to be classified. Moreover, it is a good practice not to always start the sequence from the same position (neutral), but rather from different poses and positions with respect to the camera. Moreover, it is sometimes the case that two signs have a similar start but a different end and get confused. In that case, there are several options to consider:

- first, one should increase the match minimal length
- then, the recognition threshold on the patcher should be set lower if possible, to make it more selective
- finally, more training examples can be added to better differentiate the signs and the difference between the sequences should be exaggerated.

5. Wekinator AdaBoost

When training a classifier, each data point in the sequence is considered a single training example (unlike with DTW where is it all the sequence); a few sequences only may suffice and the accuracy of the input should be set to maximum while the data rate does not matter for the recording of the training examples. During the recording sequence, the user should move his hands in all the space to make sure that the training set is diverse enough to recognize the poses at different positions.

6. Threading

Max has several internal threads, which it uses for different kinds of operations. Time critical operations are processed in the high priority thread, whereas the other operations are processed in the main thread. I observed that node.js scripts outputs are by default in the high-priority thread. I have kept the time critical data such as the motion capture data from PoseNet or HandPoset on that thread, while deferring other types of data to the main thread. However, if the user wants to connect the motion tracking data elsewhere, for instance to its own patchers, it is important to make sure that it is deferred to the main thread in order not to overload it. For additional information about threading, check Max documentation on the subject.[70]

7. Body normalization in PoseNet

The qualitative improvement of the translation- & rotation-invariant transformation was also observed during the initial tests with those 8 features by moving in space and taking slightly different orientations to the camera. However, the improvement given by the normalization of the body joints with respect to body dimensions is still to be tested with more users. From my own testing, I have had performance losses when using a model trained with only one user with another user, whose body was significantly different in size. I recommend the user to add training examples recorded by himself to ensure a good performance.

E. Learning in Soundpainting with digital tools

In this section, I would like to return to a more theoretical discussion on learning in SP and the contribution of digital tools. We will show that learning SP is traditionally a collective mimetic process which no longer exists in an individual exploration in front of a computer. Then, we will raise the importance of feedback and incentives in digital tools for learning and motivate the evolution of the program to a gamified form. Finally, I will suggest that the response of performers can be described by the cognitive load of the request and propose that computers and machines can encourage performers to expand the range of their productions.

1. Soundpainting as a collective mimetic process

To my point of view, the collective mechanisms of interactions across disciplines and inside the group are fundamental elements of learning and expanding of creativity in SP. The language allows explicitly for producing material in terms of contrasts, relations, symmetries, analogies, provocations or reactions to other contents and events, from a "blind" response to the wider range of awareness (De Peluci, 2015, page 116):

[70] See https://cycling74.com/tutorials/advanced-max-learning-about-threading.

> Some signs ask the performer to use “Blinders” – no relation with another performer...other gestures such as “Organic Development” ask the performer to really listen and make decisions based on what the other performers are doing. Soundpainting incorporates all forms of listening and watching in varying degrees - the range of listening/watching is part of the parameters of each sign. Signs such as Point to Point ask the performer to listen/watch using a very wide range awareness – either perform with “Blinders”138, make a relation to something just performed, or being performed, or offer something entirely new and unrelated to anything that has been previously performed.

The collective aspects of SP are well described in its literature (Faria 2016, Duby, 2006 and Peluci 2015, p. 114) and are shown in conjunction with its qualities for learning and teaching.

At the core of this collective awareness lie processes of attention through real-time sensory and perceptual coding, optimal attention allocation, event interpretation, decision making, prediction and anticipation, such that performers are challenged in their ability to stay aware of the collective composition while at the same time focused on the soundpainter: there is a great cognitive load both during learning phases and performances in SP.

2. Cognitive load and evolution of responses in an orchestra

I propose understanding the cognitive load inside a SP in several ways:

- the cognitive load defines the limit of complexity of the response of the performers (the heavier the cognitive load, the more performers will choose simple means of production for their response)
- the cognitive load characterizes the range of possible responses (the heavier the cognitive load, the more performers will choose prototypical responses that they already are familiar with)
- the cognitive load decreases as the performers learn the concepts of SP and get faster and better as identifying the requests
- the cognitive load increases with the velocity of the requests, i.e. the speed at which the soundpainter requests the performers to respond to his signs

Although I cannot support this theory with a statistical analysis of the influence of learning and expertise in SP on the time and variety of responses of performers, my personal observations yield that most performers start with the prototypical responses that have been shown to them and explore/example the concept further as they rehearse a content and the identification of signs. Learning can therefore be seen as the expansion of a concept with the decrease of the cognitive load for producing new material.

At the contrary of human performers, digital tools have huge processing capabilities and are able to produce very complex material in little time if they are programmed for it. Therefore, they are interesting elements to integrate to an orchestra and learn with by imitation: the relation with their different production possibilities might push the performers to new directions.

3. Digital tools as exploratory and creative instruments

From the point of view of the soundpainter, the recognition is an exploratory and creative instrument which can also push for new ways of signing and thinking SP as a direct relation to the instrument (or device) itself. Learning the sign language with a group is not offered to everyone and most groups are interested in practicing with experimented soundpainters only. In practice, most soundpainters first learn the language as performers before signing themselves, such that they have already internalized most of the language and compositional propositions before endorsing the "role" of soundpainter as a composer.

4. Learning individually: the importance of feedback and incentives

Because in collective approaches to SP, soundpainters have many forms of feedback to their messages, learning the sign language individually in front of a computer may look very uninteresting and perhaps pointless in the absence of a feedback. One could also consider that using the language outside the context that it has been designed and optimized for is already a modification of the language itself and that some of the aspects of SP can only be learned from within the collective. Clearly, my intent here is rather to provide an interesting extension of SP to machine communication and new performance tools than building a realistic learning environment at this stage. I would like to discuss here the importance and the potential of feedback and incentives in the learning process and how could the program evolve toward more interesting user experiences.

Feedback is one of the most important aspects of learning. One will for instance learn how to correct himself from errors when he will be able to perceive those as such and understand what the cause of the error is. We have seen that although there are no mistakes on the side of the performer who interprets the sign, the soundpainter can make syntactic mistakes. In a learning tool which implements a grammar, it is important to let the user know why a particular sentence is wrong or what did the program recognize that is not intended. In the actual state of my program, the user can already receive these types of feedback from the automaton, which outputs not only his actual state and how the request is created but also error messages that indicate if an unexpected or illegal sign has been observed. For instance, if the user signs “numerics, long tone, whole group, minimalism, play”, the automaton will output an error message when receiving “whole group, minimalism”, stating that numerics have already been requested a content previously in the sentence and that the request of a new content is ambiguous. Another type of feedback is the ability to hear the contents that are produced by the program and how they react to the different requests, even when the user is making mistakes or the program doesn’t recognize the intended signs.

Feedback is one form of incentive to explore more of the tool and learn with it. They can be classified in two categories: external or internal to the program. Some artists are already interested in using the recognition tool in their own installations: they have external incentives. However, some users may not be familiar with digital tools nor with SP and will not take to create something of their own if they are not pushed to it by the program’s mechanics that form its internal incentives. Typically, games are good examples of programs with a lot of internal incentives. They have so-called gamified features, such as a score, elements of competition or collaboration, rewards, etc, which push the user to exploring more of the game and performing better at it.

I think that there is a great potential to be explored in my tool with the implementation of gamified elements or feedback through interactive designs and visualizations, for instance on the model of the V motion project. While some artists are already (and hopefully more will be) interested in my work because of what it offers for their own work, I would like to reach a broader public in the future, perhaps outside the Soundpainting, Max/MSP or media artists communities. By reworking the design in the form of a game, it would be possible to invite people such as amateur musicians, young dancers or simply “tech curious” to compose in real time with built-in elements on using their notebook, tablet or phone, while by polishing the customization pipeline and the extent of devices that can be connected to it, I hope that experienced artists could appropriate themselves the tool and create new forms of manipulation of sound through movements and gestures.

F. The future

1. Short and mid-term plans

a) A compiled release

The first step that I would take is releasing a compiled, standalone app of the recognition program for both Windows and Max OS. This would allow for a broader distribution of the tool outside the Max/MSP community and a less painful installation process. However, there are some challenges to the compilation process that I could not resolve yet, specifically dynamically resolved file paths in the patcher that are broken after the compilation.

b) Beta testing and performance assessment

Once the program will be released, I plan to reach beta testers (a few people already were interested) to make a more collaborative and community-oriented development plan. During the beta testing, it would also be possible to make the qualitative and quantitative performance assessments that I could not make during my master thesis:

- testing on a multi-user training database
- classification performance evaluation (PoseNet and HandPose) for N>10 experienced soundpainters: reactivity, confusion matrix and settings optimization
- unsupervised usability testing among people that are
    - both experienced in SP and digital performance tools
    - experienced in SP but not in digital tools
    - experienced in digital performance tools but not SP
    - not experienced in either
- assessing the performance of collaborative versus individual training database

c) Sign database and dictionary

The work with beta testers should also provide us a collaborative sign database that could be integrated in Thompson's dictionary, both in video format but also in motion tracking normalized format. A common workflow could perhaps be designed with the dictionary team in that sense. I also hope that my contribution to understanding the SP grammar could lead to future research and help structuring the dictionary.

d) Collaboration leads

During the master thesis, I had the opportunity of sharing my demo videos in both SP and Max/MSP communities on the social networks. I am very glad that the reception of my work was very positive and got me in touch already with a few artists, students and teacher from all over the world. While I do not have concrete ideas on what this work will lead me to, I wish that these first contacts could bring interesting follow-ups and that I will later be able to continue this work on a more collaborative and artistic side.

e) Cargo Bike Band tour

As mentioned in my initial calendar, I plan to use the tool for the first time in public for another project of mine, the Cargo Bike Band. Although our tour plans got greatly reduced by the COVID-19 pandemia, we are still planning on using our itinerant music and video setup prototype around Geneva's lake for a few weeks and will experiment with the sign recognition tool during our performances. I plan to test the setup on the bike right after my defense on the 3rd of July 2020.

2. Improvements, projects and potentials for the long-term

f) Bach Project: meet Cage and Dada

The Bach project comprises 3 libraries: Bach, Cage and Dada. Although I am only using Bach yet for implementing previously defined contents, Cage brings the power of generative models for music into the game. While it requires me some time to implement this, it would end up simulating much better the reactions of an actual orchestra of performers and allow for interacting generative models in which one virtual instrument can react to the others. Dada adds an additional level of potentialities by bringing non-conventional representations of music in graphic, ludic and explorative approaches.

g) Hi5 gloves

Although I have not been to explore hand recognition with the Hi5 gloves yet, I hope to do so in the future. At the difference of HandPose, it would allow the user to move further in space and to manage occlusions much better (for instance for the "extended techniques" signs, one hand lies in front of the other and HandPose typically performs very bad at identifying the two hands). It is clear for me that HandPose cannot provide equivalent results to motion tracking gloves and highly restrict the position of the user and space. Moreover, while I am confident that HandPose will soon update to multiple hands tracking[71] , only the gloves can consistently recognize both hands at all time, should they be occluded by other body parts or objects. Building a dedicated C++ object from the SDK is the most practical way to add them to the system, while it would also bring an interesting input to the whole Max/MSP community.

[71] See https://google.github.io/mediapipe/solutions/hands#with-multi-hand-support

h) FaceMesh and Modosc: new sets of features

Although we discussed the choice of meaningful features for recognizing SP signs within the default SP mode, I would like to discuss the “Modosc”[72] motion descriptors library for Max/MSP and FaceMesh as other interesting inputs.

In the “shapeline” mode, all gestures and signs made by the soundpainter (except the “exit” sign of the mode) are interpreted in a figurative way by the performers, i.e. in an iconic or suggestive way rather than in a symbolic way. For instance, the soundpainter could use his facial expressions to convey emotional content or imitate the throwing of a virtual ball in the space and let the performers interpret (abstractly and freely) the dynamics of the scene. Interpretation in such a mode, as we have seen previously, involves particular cultural knowledges as well as knowledges about emotions, facial expressions… whether they are cultural or not.

The EyesWeb project (DIBRIS - University of Genoa, s.d.) proposes several “expressive cues” for analyzing body movement and gestures in relation with their emotional content and creating models of interaction between gestures and musical languages, in analogy to what the mode “shapeline” offers in SP. Some of these cues are features such as the contraction index, fluidity, curvature of a movement, jerk (first derivative of the acceleration), symmetry with respect to different points of the body, directivity, etc.

Although the authors of this system are not directly demonstrating that these are the features the cognitive system uses for triggering emotions (Piana, Stagliano, Odone, Verri, & Camurri, 2014), the experiments and interactive performances they present suggest that these features do capture some sense of the emotions conveyed through gestures.

In Max, the “Modosc” library allows for computing of some of these cues. In future extensions of the project, it would be relevant to explore simple interpretations of emotional contents through gestures with music or visuals contents in the shapeline mode based on such features for PoseNet, even though more performant models would probably come from non-interpretable machine learning models in the near future.

[72] Modosc can be found at https://github.com/motiondescriptors/modosc, with its relevant literature.

FaceMesh is a Tensorflow package released in March 2020[73] that infers approximate 3D facial surface geometry from an image or video stream and that can be ported to Max just as easily as PoseNet or HandPose models. One use of this package for SP could be building a lightweight expression and emotion classifier that could also be used in SP modes such as the shapeline. There are already several convincing attempts at recognizing facial emotions but FaceMesh advantage is that it opens the way for fast, real-time emotion recognition from 3D mesh, hence independent of the user's face color, dimensions, eyebrow shape, etc. Just like a normalized skeleton from PoseNet allows us to build a simple yet efficient model for recognizing SP signs without heavy training sets and models, the FaceMesh could let us recognize emotions as facial signs… but also creating our own, new signs with the face.

i) From automaton to Conditional Random Field or Hidden Markov Models with Viterbi algorithm

Automata are quite simple abstract machines, that are efficient to parse grammars and easy to integrate in such a project. However, there are some strong assumptions behind the use of an automaton:

1. it assumes that all the grammatical rules are known explicitly (not only internalized) by the programmer
2. it assumes that the signs that are given to it as input are 100% right

In my recognition tool, none of these are true. The first assumption can be taken down by using Hidden Markov Models (HMMs) or Conditional Random Fields (CRFs) to parse the grammar. Instead of being programmed, these algorithms learn the grammar from many examples of sequences in which words are labeled by their syntactic categories and functions. CRFs are much more general than HMMs and while I cannot get in further details, they allow for modelling context sensitivity at higher order than HHMs; they can be thought as a generalization of HMMs.

Both HMMs and CRFs can also take down the second assumption by using linear inference with the Viterbi algorithm. This algorithm was introduced by Andrew Viterbi in 1967, originally to find the most likely sequence in a set of probabilistic observations. As an exemple, imagine that the recognition tool receives the following data from Wekinator:

I. Whole group (probability = 0.695) or Rest of the group (probability = 0.214) …
II. Long tone (probability = 0.943) …

[73] See https://blog.tensorflow.org/2020/03/face-and-hand-tracking-in-browser-with-mediapipe-and-tensorflowjs.html.

III. Brass (probability = 0.385) or Extended techniques (probability = 0.601) …
IV. Minimalism (probability = 0.906) or Maximalism (probability = 0.082) …
V. Play (probability = 0.804) or Whole group (probability = 0.112) …

Here, the index 1, 2, …, 5 represent each successive observation. At each observation, it is not only one sign that is identified, but a probability distribution of signs from Wekinator. For now, the system only considers a sign once its probability is higher than a threshold, for as long as another sign is not recognized. It would recognize "Whole group, Long tone, Play" only and would identify "Extended techniques" and "Minimalism" as grammatical mistakes.

But with CRFs and HMMs, it would be possible to have much lower probability thresholds and sort out false positives by looking at the whole sequence instead of signs individually. The Viterbi algorithm basically maximizes the likelihood of the sentences and can be viewed as an "error correcting" method - it is able to correct the misclassification of one observation based on the whole sentence. In our example, it would recognize "Whole group, Long tone, Brass, Minimalism, Play" and correct the misidentification at step 3.

All CRFs do not implement the Viterbi algorithm. I have been advised to look at Wapiti[74], however I would highly prefer a Java, Node.JS or JS script to be able to port it into Max. I have found a python wrapper for Wapiti[75], but the compatibility of Max with python is experimental[76] and I haven't investigated it yet.

j) OSSIA Score and multimedia connectivity

Even though I decided to drop OSSIA Score for now, I would still consider it as an important tool for the future and keep an eye open on its evolution in the following months and years.

k) Interface and userflow

The interface and userflow can be improved much more in the future evolution of the tool. The graphic possibilities for creating interfaces in Max are huge but getting passed the integrated options require a great amount of time and involve objects of different

[74] See https://wapiti.limsi.fr/.

[75] See https://github.com/adsva/python-wapiti.

[76] See https://github.com/grrrr/py.

nature (“jsui” objects[77]). Typically, the tool could evolve with menus, more dynamic elements, buttons and a polished, design look. We could also make a step towards VR or think about its use with a gamified interface, like the V motion project.

l) PoseNet and HandPose prediction filters

PoseNet and HandPose sometimes behave in a strange manner when the accuracy of recognition is very low: one joint can be recognized at a non-realistic place and the data flow have non-continuous “jump” during which a joint can move at unrealistic speeds from one point to the other. In general, it is best to configure the setup so that the accuracy is high enough for this situation never to happen; but in the case of an occlusion, poor lightning environment, etc, it would be nice to implement prediction filters in PoseNet and HandPose directly to omit unrealistic jumps and joints positions. Kalman filters[78] are typical choices for comparing realistic predictions to the data and cleaning the erroneous data.

m) Input consensus

The idea of having several inputs made me think a lot about constructing some kind of consensus between different motion tracking methods, in order to increase the spatial range of the tool, its use cases and genericity. Several consensus methods already exist, and it would be a challenging yet very interesting lead for augmenting the tool’s performance.

n) Communication with other performers

If we think of the computer as a performer in the orchestra, one need to take into account not only the communication with the soundpainter but also the whole artistic production of the orchestra. Typically, the computer could be linked to microphones and cameras that allow it the capture and identify (with a separate classification system) what the other performers are doing and interact with them. For instance, in the actual state of my program, it would be impossible to implement the content “synchronize” or “synchronize with” + identifier, which refer to the production of other performers. However, there are already many systems in music and graphics that could help represent the interactions between performers: beat detection, style/genre, harmony and melody identification, stylistic accompaniment or generative improvisation models, etc.

[77] See https://docs.cycling74.com/max8/refpages/jsui.

[78] See https://en.wikipedia.org/wiki/Kalman_filter.

Combining those with the recognition of the sign language would be a heavy work but nonetheless an important direction to take for bringing computers onto the stage.

o) VR, visualization system and user feedback

Adding immersive features and visualization systems to the tool would greatly improve its qualities both for learning and performing. For instance, the learner could be able to reproduce gestures based on an avatar in a virtual 3D space or a screen. For performing, just like the V motion project, gamified visualizations would highly add value to the performance and its preparation.

So far, the motion tracking points are only displayed in the PoseNet and HandPose video feedback screen. One could first easily display them in a jit.world object in Max and render it on a separate panel.

Another idea would also be to link the state of the automaton with the video feedback, such that a transition and the end of sentences would be illustrated with interactive (and possibly immersive) visuals in real-time.

Finally, I have presented the orchestra simulation as a sound simulation only, but one could also visualize a long tone graphically and create interesting links between sound and video in the performance, such as a score, the interpretation of a content in terms of graphics or a visualization of the sounds directly.

p) Other Soundpainting modes and non-symbolic gestures

To incorporate additional modes into the recognition tool, the automaton would need to be updated and the set of inputs completed to account for non-symbolic gestures that are not defined in the SP dictionary: icons (hits, soft movements) or indices (facial emotions, body expression). One possibility would be to collaborate with existing projects which are already focused on these other types of gestures such as GeKiPe.

This is clearly a very long-term work that implies implementing many types of signs and grammars that are found in SP, for which the actual inputs and automaton do not suffice. Clearly, the recognition tool will never match the complexity of SP but it will be interesting to think about what modes and input would add value to the performance with computer music.

q) Gamification, mobile support and reaching a wide public

Finally, in the very long term, I imagine that the tool could incorporate a lot of gamification mechanisms and operate on several devices, from phones, tablets to high-tech computer rigs. There are already a few games that are structured in several "modes"[79]:

- a mode for the wide public in which you can play the game, compare your performance to other people and learn in the process
- a mode for creators in which people can create their environment, customize the mechanics to fit their needs and then play with it and share it to other people
- a very low-level developer access, which allows for programming new parts and mechanics of the program itself

I could easily imagine such a structure for a "gamified" artistic sign language game, very far from my own proposal with SP yet, but nonetheless inspirational for the continuation of this work.

[79] I am thinking about games such as trackmania, skyrim and the skyblivion, skywind projects.

III. Conclusion

This master thesis offers a double perspective: building a model of Soundpainting from the point of a view of linguistic and implementing it in a recognition program. I have first approached Soundpainting in a broad historical and theoretical context that permitted us to analyze its language in a structuralist approach. We saw how it can indeed be decomposed at several levels and treated as abstract mechanisms of production of signs, sentences and meaning. After looking at the origins and types of signs themselves, I provided a deeper insight into its grammar by re-defining the different lexicon and syntactic categories, their functions and the hierarchical structure they form. We also pointed out elements of morphology, describing how several signs or morphemes combine into other, more complex ones. The description of this other part of the grammar is usually forgotten and like other sign languages, suffers from the high complexity of its different sequential and simultaneous structures. In a broader perspective, we saw that the performativity of the language was partially due to its ability to mix up a variety of concepts and push their limits to invite performers at creating, new, interesting material.

The limits of this simplified model were soon reached by considering configurations that embody compositional choices, default parameters, roles in the communication and other contextual elements. However, it has clarified several ambiguities and confusions in the usual introductions to Soundpainting by pushing for a clearer separation of grammatical rules and configuration choices such as compositional rules. I discussed how the latter were incorporated in its definition from its historical development, diffusion and adoption mechanisms. To conclude, I suggested that it can only be understood by considering both the linguistic perspective and the practices that are commonly identified as Soundpainting.

In the second part of the report, I presented a prototype of Soundpainting recognition program made with Max/MSP and Wekinator, which allows the users to create, train and classify their own signs within a high-level user interface. In the recent years, several projects have been developed for music and sound generation through movements and two series of attempts were already made on the side of Soundpainting recognition. I discussed some of their limitations and provided a lightweight system that overcome most issues and offers a large potential of future extensions.

By the creation of this tool, I wanted to target specifically the Soundpainting community and in a broader sense, a public of non-programmer artists. This led me to build a simple and generic interface that could be both used “as such” or customized by the expert users.

The simultaneous development of both the theoretical model and the tool resulted in a similar structure. Indeed, I proposed an implementation in several independent layers, each performing a specific function, from the creation of a sign to the grammar parsing of their temporal sequence. In fact, both the theoretical and practical parts of this report strongly influenced and explained each other, such that they should be considered as complementary.

At the global level, the program can simulate a “simple” Soundpainting orchestra and recognize at least more than 20 gestures performed at both full body and hands scales. Although in this first release, only the internal orchestra simulation and the integration with Ableton Live are implemented, it offers a generic OSC output that allow simple integration with additional environment, devices or programs such as OSSIA Score. Therefore, there are endless possibilities for creating interactive instruments within existing Soundpainting orchestras or multimedia installations. Moreover, I have designed the tool so that it could be (later) used as a learning tool, in collaboration with the building of a dictionary by and for the Soundpainting community.

This work-in-progress assembles very different existing pieces of technology to form an unprecedented creative human-machine interface. The early interests of several artists and programmers and the large potential offered for its new developments are likely to build a bright and promising future in the field of real-time music composition.

IV. Bibliography

La Percumotora. (s.d.). *Ritmo y Percusión con Señas*. Récupéré sur La Percumotora: https://www.lapercumotora.com/lapercumotora/ritmo-y-percusion-con-senas/

Aerts, Broekaert, Gabora, & Sozzo. (2016). Generalizing Prototype Theory: A Formal Quantum Framework. *Frontiers in Psychology*.

All About Jazz. (s.d.). *Butch Morris* . Récupéré sur All About Jazz: https://www.allaboutjazz.com/butch-morris-butch-morris-by-aaj-staff.php

Barakat, R. (1975). *Cistercian sign language: A study in non-verbal communication.*

Brock, A. (s.d.). *Semantics #1 - Signs and Meaning in Language*. Récupéré sur Youtube: https://www.youtube.com/watch?v=h2PDhtqDgKg

Brock, A. (s.d.). *Semantics #4 - Prototype Theory*. Récupéré sur Youtube: https://www.youtube.com/watch?v=mff_sPnz_gs

Carmona J.M., C. J. (2012). A Performance Evaluation of HMM and DTW for Gesture Recognition. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications.*

Choi, H.-r. &.-Y. (2017). Directional Dynamic Time Warping for Gesture Recognition. 22-25.

Couture, N., Bottecchia, Chaumette, Cecconello, & Rekalde, e. a. (2017). Using the Soundpainting Language to Fly a Swarm of Drones. *Advances in Intelligent Systems and Computing*.

Dau, Mugnier, & Stumme. (2005). Conceptual Structures: Common Semantics for Sharing Knowledge. *International Conference on Conceptual Structures*, (pp. 102-104).

DecodingScience. (s.d.). *Ferdinand de Saussure: The Linguistic Unit – Sign, Signified and Signifier Explained*. Récupéré sur Decode Science: https://www.decodedscience.org/ferdinand-de-saussure-the-linguistic-unit-sign-signified-and-signifier-explained/

DIBRIS - University of Genoa. (s.d.). *The EyesWeb Project*. Récupéré sur Infomus: http://www.infomus.org/eyesweb_eng.php

Duby, M. (2006). *Soundpainting as a system for the collaborative creation of music in performance.*

Duchez, M.-E. (1979). La repŕesentation spatio-verticale du caractère musical grave-aigu et l'élaboration de lanotion de hauteur de son dans la conscience musicale occidentale. Dans I. M. Society, *Acta Musicologica, Vol. 51, Fasc. 1*.

Faria, B. (2016). Exercising musicianship anew through soundpainting: Speaking music through sound gestures.

Guyot, P., & Pellegrini, T. (2016). Vers la transcription automatique de gestes du Soundpainting pour l'analyse de performances interactives. *Journées d'Informatique Musicale*.

Hiyadi, H. &. (2016). Combination of HMM and DTW for 3D Dynamic Gesture Recognition Using Depth Only.

Hofstadter, D. (2009). *Analogy as the Core of Cognition.* Récupéré sur Youtube: https://www.youtube.com/watch?v=n8m7lFQ3njk

Hopcroft, J. E., & Ullman, J. D. (1979). Introduction to Automata Theory, Languages, and Computation.

Huglo, M. (1963). La chironomie médiévale. *Revue de musicologie*.

Jáuregui, G., Dongo, & Couture. (2019). Automatic recognition of Soundpainting for the Generation of Electronic Music Sounds.

Liddell, Henry, G., & R., S. (1940). *A Greek-English Lexicon*.

Minors, H. J., & Thompson, W. (2012). *Soundpainting blog*. Récupéré sur Soundpainting: http://www.soundpainting.com/blog/

Pâris, A. (s.d.). *Direction d'orchestre.* Récupéré sur Universalis: https://www.universalis.fr/encyclopedie/direction-d-orchestre/

Peirce, C. S. (1903). *Nomenclature and Division of Triadic Relations, as far as they are determined.*

Pellegrini, T., Guyot, P., Angles, B., Mollaret, C., & Mangou, C. (2014). Towards soundpainting gesture recognition.

Peluci de Castro, G. (2015). *Problemas de performance em improvisação dirigida.*

Piana, Stagliano, Odone, Verri, & Camurri. (2014). Real-time Automatic Emotion Recognition from Body.

Quay, S. (2001). Signs of Silence: Two Examples of Trappist Sign Language in the Far East. *Cîteaux: Commentarii cistercienses, Vol.* 52 *(3-4)*, 211-230.

Raheja, J. &. (2015). Robust gesture recognition using Kinect: A comparison between DTW and HMM. *Optik - International Journal for Light and Electron Optics*.

Revlin, R. (s.d.). *Cognition: Theory and Practice.*

Rosch, E. H. (1973). Natural Categories. *Cognitive Psychology 4*, 328-350.

Salvador, S. (2004). FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space. *Intelligent Data Analysis*.

Shannon. (1948). *A Mathematical Theory of Communication* .

Thompson. (s.d.). *Soundpainting*. Récupéré sur Soundpainting.com: http://www.soundpainting.com/soundpainting/

Yerlikaya, & Coskumer. (2016). Analysis of Soundpainting sign language visuals. *MĀKSLA UN MŪZIKA KULTŪRAS DISKURSĀ.*