PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

DESIGN OF A MAPPING STRATEGY FOR COMPUTER SOUND SYNTHESIS IN MULTI-TOUCH SURFACES BASED ON EXPRESSIVE ANALYSIS OF GESTURES

MARIE CARMEN GONZÁLEZ INOSTROZA

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
RODRIGO CÁDIZ

Santiago de Chile, October 2018

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

DESIGN OF A MAPPING STRATEGY FOR COMPUTER SOUND SYNTHESIS IN MULTI-TOUCH SURFACES BASED ON EXPRESSIVE ANALYSIS OF GESTURES

MARIE CARMEN GONZÁLEZ INOSTROZA

Members of the Committee:
RODRIGO CÁDIZ
DENIS PARRA
JAVIER JAIMOVICH
LUCIANO CHIANG

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, October 2018

Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.

Ada Lovelace

ACKNOWLEDGEMENTS

First and foremost, I have to recognize my research supervisor, Prof. Rodrigo Cádiz. Thanks for his help and guidance in all this years since I started as an undergraduate researcher. I specially appreciate his encouragement to participate in different academic processes and his passion for interdisciplinary research.

Thanks to all the people who have been part of the Center for Research in Audio Technologies. Specially to Prof. Patricio de la Cuadra, whose advice made me think better, and to Aaron, Constanza, Andrés, Gabriel, Veronica, Vernon, Danae and Domingo, who have been part of the discussions of this work. I also wish to thank Prof. Alvaro Sylleros and the members of Ludique for their valuables remarks about the project.

My research would have been impossible without the aid and support of my family and friends. Thanks to my parents, Claudia and Francisco, who taught me to love engineering and never give up. To my siblings, Claudia and Francisco, whose questions and curiosity challenge me every day.

Special thanks to my boyfriend, Andrés, whose love and support let me overcame bad days and enjoy sunny ones. For staying with me when nothing was working and make me believe that everything will be ok.

I am profoundly grateful to the support of my friends Pablo, Coni, Paula, Champi, Clau, Rorro, Soto and Kike. We have shared dreams and nightmares about electronics and technology. I wish we follow developing and dreaming together. Also thanks to my life friends Camila, Renata, Nayareth and Nataly, who make my life happier. Besides, thanks to Erick Poulsen, Matías Bustos and Marcelo Pizarro, who had discussed with me about music and research.

I really appreciate the opportunity to be part of some organizations in my period at the University. First, to the student branch IEEE UC. Thanks to the old members, Nacho, Wladi, Sergio, Rorro, Nico, Pancho, Diego, Guille and Braulio, who were my real teachers in different fields. Also to all the guys with whom I have shared the last years: Simón, Renzo, Kevin, Fernando, Timmy, Dedes, Kari, Guali, Dani, Andrés, Alfaro, Rodo, Jacoby, Kike. Also thanks to the robotic's students team ICARO, which was a space of continuous challenge and learning. Finally, to the folkloric team Almas UC, that let me enjoy dancing and develop self-confidence.

To all the Department of Electrical Engineering for being an space of growing and my second home. Specially thanks to the staff, Karina, Tía Vicky, Don Carlos, Mary and Karina for receiving me every day with an smile. Also thanks to the members of IC-UC, who have shared with me as part of the team, specially to Jorge, Renzo, Matías, Nawel, Juan Andrés and Prof. Angel Abusleme.

I also would like to express my admiration to some people who made this work possible. To Ada Lovelace, Marie Curie and Alan Turing, who have inspired me as researcher to work strongly and open-mindedly. To the musicians that constitute the soundtrack of this thesis and who made me love music, specially to Los Bunkers, Ases Falsos, Los Tres, De Saloon, Kuervos del Sur, Soda Stereo, Los Prisioneros, The Beatles, The Doors, The Clash, The Ramones, Aerosmith, The Strokes, Franz Ferdinand, Arctic Monkeys and Jack White. Also thanks to the authors whose lines make me dream with a better world (probably one where most of their forecasts aren't true): Isaac Asimov, Justin Cronin, H.G. Wells, Gabriel García Márquez, Roberto Bolaño, George Orwell, J.K.Rowling, Ray Bradbury, Haruki Murakami, Orson Scott Card, Julio Cortazar and Stephen King.

This research was partially funded by the National Commission for Science and Technology Research of Chile (CONICYT), under grant Fondecyt #1161328, and Consejo Nacional de la Cultura y las Artes, Government of Chile, under grant Fondart #219276. A very special gratitude goes out to School of Engineering UC for partially funding my participation in the ICMC 2017 conference to present part of this thesis, and my internship at INRIA-Bordeaux. Also thanks to POTIOC team and INRIA-Chile, for all their help in my internship and my academic doubts.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

RESUMEN

Como parte del proyecto Arcontinuo, un instrumento musical digital (DMI) con una superficie multitáctil curva, esta investigación busca diseñar una estrategia para una síntesis de sonidos expresiva. Para lograrlo, este estudio contempla dos partes: análisis de la interacción en DMIs con superficies multitáctiles y una propuesta de mapeo de descriptores expresivos gestuales basado en lógica difusa.

La calidad de la interacción debe ser un objetivo central en el diseño de un DMI, pero no es una tarea simple ya que involucra relaciones entre el músico, el DMI, otros músicos y la audiencia. Por ello, usando conceptos de Interacción Humano-Computador (HCI) y la teoría de *affordance*, se propone un modelo que permitiría iluminar el diseño de eventos de interacción en DMIs multitáctiles. Los músicos deben interactuar con interfaces que pueden ser controladas con *frameworks* de interacción y procesamiento, por lo que es importante pensar y diseñar qué ocurre en estas etapas y cómo se relacionan con el mapeo y la síntesis de sonido. Por lo tanto, debemos asegurar que el uso de los *frameworks* permita el control de la síntesis para tareas específicas, como la interpretación musical o la mezcla, de acuerdo con los deseos del músico.

La incorporación de un análisis numérico de los gestos en base a la expresividad podría proveer de una etapa de mapeo más rico. Algunos de los descriptores más reportados en la literatura y los de esfuerzo y forma de Laban han sido utilizados. Este mapeo enriquecido es completado con un sistema de control difuso que recibe los descriptores como entradas y los mapea en variables de síntesis. Esta herramienta permite a los diseñadores de DMIs definir sus propias reglas de mapeo considerando descripciones expresivas de los gestos con diferentes estrategias: basadas en descriptores, basadas en un conjunto de gestos o basadas en metáforas musicales.

Palabras claves: Control gestual, Instrumento Musical Digital, multitáctil, expresividad.

ABSTRACT

This research is part of the project Arcontinuo, a digital musical instrument (DMI) with a curved multi-touch surface. This work focused in the design of a strategy that would let us synthesize sounds in an expressive way. To achieve that, the research was divided in two parts: analysis of interaction in DMIs with multi-touch surfaces and a proposal for mapping of expressive gestural descriptors based on fuzzy logic in multi-touch surfaces.

Interaction design for DMIs is not a simple task because it involves many relations between the musician, the DMI, other musicians and the audience. The interaction quality must be a central objective in a DMI design process. Using concepts from the field of HCI and affordance theory, we propose a model that could illuminate the designing of interaction events in multi-touch DMIs. Musicians should interact with interfaces that can be controlled with interaction and processing frameworks, and as such, it is important to think and design what happens in these stages and how they relate to mapping and sound synthesis. Therefore, we must ensure that DMIs use frameworks that allow users to control the synthesis process for specific tasks, such as musical performing or mixing, in an appropriate way in accordance with their desires.

The incorporation of a numerical analysis of gestures based on expressiveness may provide a richer mapping stage. Some of the most used descriptors that have been reported in the literature and Laban's descriptors for effort and shape were used. This enriched mapping is completed with a fuzzy control system that receives the descriptors as inputs and maps them into synthesis variables. This tool would allow DMI designers to define their own mapping rules considering expressive gestural descriptions with different strategies: based on descriptors, based on a subset of gestures or based on musical metaphors.

Keywords: Gestural control, Digital Musical Instrument, multi-touch, expressiveness.

CHAPTER 1. INTRODUCTION

Figure 1.1. Second Prototype of the Arcontinuo digital musical interface. It is a musical instrument with a curved multi-touch surface.

The digital synthesis of sounds, with the help of computational systems, has expanded the realm of possible sounds that musicians can use. Acoustical and electronic traditional musical instruments are physically limited by their sound production technique, restricted by the laws of physics. Instead, computational sound synthesis allows the creation of any sound, even the ones that aren't possible in nature due to spatial or material limitations (Marshall, 2009).

The desire to control sound synthesis on stage has brought the creation of a variety of digital musical instruments (DMIs). These instruments can generate any sound from any movement that the musician can do (Paradiso & O'Modhrain, 2003). Traditionally, the design of DMIs has not considered all of the musicians requirements and their necessities on stage. For example, ergonomics features of instruments and movements are not always carefully designed. Also, the quality in the interaction between musician and DMIs has not been study in enough depth (Sylleros et al., 2017).

With these ideas in mind, the musical instrument Arcontinuo was designed: an ergonomic DMI with a centered-on-the-musician design methodology. What movements and postures the musicians can do were studied in detail to decide the shape; the instrument has a touchable curved surface that can be hanged from the shoulders, as figure 1.1 shows. The surface measures the pressure that the musician exerts with its fingers and enables them to perform with natural and ergonomic movements (Arcontinuo, 2015; Sylleros et al., 2014).

As part of the project Arcontinuo, this research focused on multi-touch surfaces and how to play music with them. As it was exposed by (R. F. Cádiz & Sylleros, 2017), it was decided to give users the chance to change the way they interact with the instrument and take decisions about the way it works. Nevertheless, it is necessary to provide a solid foundation that would help musicians to design interactions that can work properly in any required context.

1.1. Hypothesis and Objectives

The central hypothesis of this work is the following: A processing of gestural data based on expressiveness allows for mappings that bring expressiveness to multi-touch digital musical instruments.

Following this hypothesis, the main objective of the study was the development of a system that would allow the fine control of sound synthesis based on fingers' gestures on a multi-touch surface with a particular focus on expressiveness. We have implemented and tested our approach with the instrument Arcontinuo, but it is general enough to be applied for any musical instrument based on continuous touchable surfaces.

In order to achieve the main objective, the following specific objectives were raised:

- Implementation of an algorithm that allows for the detection of fingers in the Arcontinuo fast and accurately
- Analysis of interaction in different digital musical instruments based on multi-touch surfaces
- Determination of advantages and disadvantages of different current strategies in DMIs with touchable surfaces
- Implementation of an analytic system for finger gestures in an expressive way
- Implementation of a mapping strategy from finger features to sound synthesis parameters

1.2. Literature review

As technology has evolved with DMIs, interfaces have incorporated more options for expression in time. The first versions of DMIs were based on knobs and buttons. Then, "expressive controllers became sidelined, and the market was dominated by the simple on-off diatonic organ manual, perhaps with the addition of a couple of wheels for the left hand and a pedal or two" (Paradiso & O'Modhrain, 2003). The MIDI protocol allowed the separation of the process of generation and sound control, so more sophisticated controllers were incorporated. Nowadays, the usage of Human-Computer interfaces based on multiple sensors allows for the capturing of varied physical expressions and use them to control sound synthesis engines (Paradiso & O'Modhrain, 2003).

Typically, DMIs are been understood as a three-stage process as shown in figure 2.1 (Wanderley & Orio, 2002; Wessel & Wright, 2002). This model divides the process on three stages: input controller, mapping and synthesis. The first one represents the physical

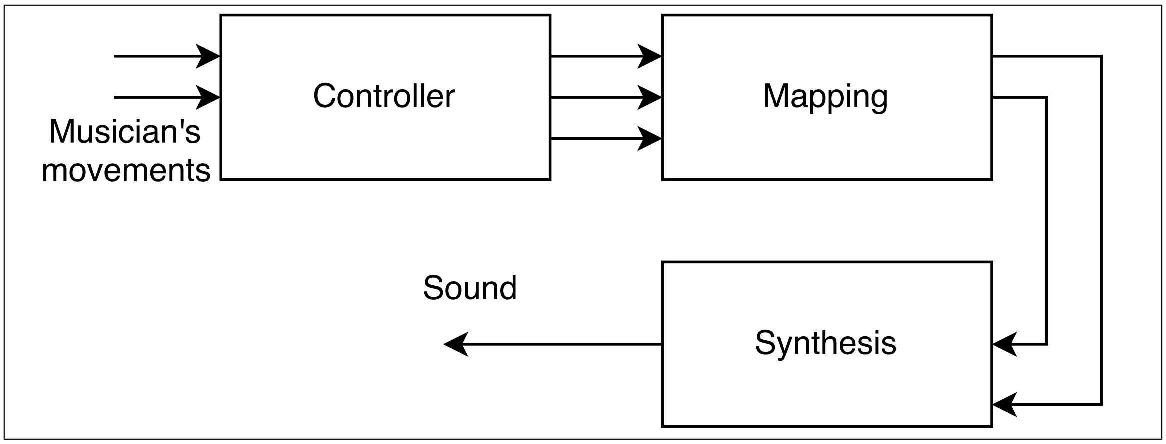


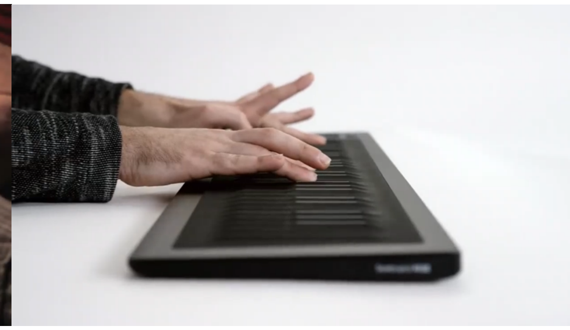
Figure 1.2. The classical model of DMIs incorporates controller, mapping and synthesis stages (Wanderley & Orio, 2002; Wessel & Wright, 2002). The controller stage does not incorporate information about subdivisions, interface controls or signal processing sub-stages that would allow the understanding of what the user is doing.

interaction and inputs for the system. The second stage translate the musician's movements into musical features or parameters that are the inputs for the synthesis stage, which generates sound. In consequence, this model allow us to divide the design of DMIs on modular and semi-independent processes, but it does not include specific details about the interaction between musician and instrument.

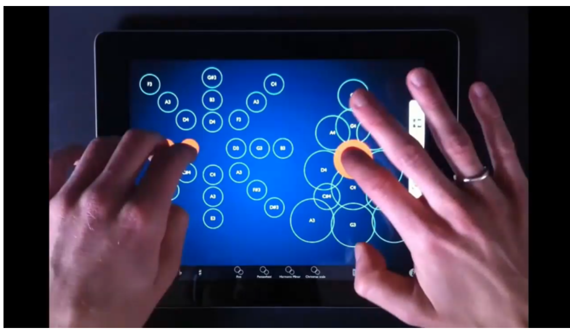
This research focuses on the specific case of DMIs that have controllers based on multi-touch surfaces and how to map musicians' gestures into sound in this case. Multi-touch surfaces are widely common in DMIs and they are incorporated in dedicated or multi-proposal devices, as figure 1.3 illustrates. More common approximations to use these technologies are buttons, keys and grids that represent a particular frequency or filters; when the musician taps an specific feature is set, so other movements are ignored (Alessandro et al., 2015; Haken et al., 1998; Trump & Bullock, 2014; Wang, 2011) This approximation does not take advantage of gestures made by the musician, neither relations between gestures in time and space.

Gestures can be defined as observed movements of the body that contain information. The analysis of gestures in computational contexts does not only require information about

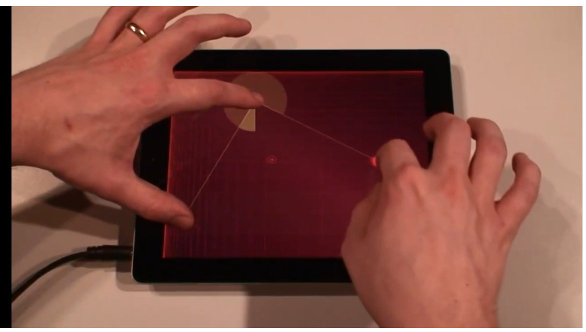
(a) Continuum



(b) Roli



(c) Orphion



(d) TC-11

Figure 1.3. Examples of DMIs based on multi-touch surfaces. The Continuum (Haken et al., 1998) and Roli (Roli, 2018) have dedicated surfaces especially designed for musical purposes. The Orphion(Trump & Bullock, 2014) and TC-11 (Schlei, 2012), in contrast, are applications that work for a generic tablet such as the iPad.

spatial position and its evolution in time (Cadoz & Wanderley, 2000) but also contextual information that allows the interpretation of meaning. These different sources of information should be catchable by sensors and processable and understandable by an algorithm (Schumacher et al., 2016; Camurri et al., 2005). In this case, just the fingers that are touching the surface can be sensed, so we require information about how they move and what it mean in the context of the interaction between the musician and the instrument.

In musical contexts, gestures have had a central role along history. In the case of traditional musical instruments, they must be mastered by musicians in order to obtain the desired sounds. In addition, performers use gestures with the intention of communicating with the audience and other musicians or as a response to the sound, and also there are gestures that facilitate the realization of sound-producing gestures (Dahl et al., 2010). This research focuses on gestures that generate and modify sounds.

Research in music and Human Computer Interaction (HCI), known as Music Interaction, has contributed to a deeper understanding of DMIs. HCI provides tools to analyze musicians' activities and to evaluate DMIs as interfaces that control a computational process (Holland et al., 2013). In HCI, *mental models* are the user's explanations about how a system works, which may be constructed by *metaphors* and modified by usage. Metaphors are representations of a system that define rules of its behavior and can refer a well-known system in order to facilitate the understanding of a new system performance(Wickens et al., 1998). On the other hand, HCI also considers *frameworks*, which are conceptual or physical systems that structure a process or a system with the aim of improving the interaction (Mooney, 2010).

As it was proposed in (Jordá et al., 2010), touchable surfaces bring actions from the physical world to a virtual representation, meaning that the usage of real-world metaphors from the users' movements and actions can provide easier ways to use the interface. However, in the case of DMIs, metaphors have been in most cases centered on mapping and synthesis, because they are considered the core of the instrument (Magnusson, 2010). In chapter 2 we argue that a more careful DMI analysis and design of the interface can give a better understanding of the system and incorporate the users' movements in a more natural way.

On the other hand, a central issue on the design of DMI is how to make them expressive. Some authors consider that expressiveness focuses on the message the composer wants to transmit, which can be modified by the performer with changes in sound (De Poli, 2004). From this point of view, gestures must facilitate expression, giving performers the ability to control sound features, producing an expressive sound (Arfib et al., 2005). As "behaviors encode content information (the 'What' is communicating) and expressive information (the 'How' it is communicating)" (Pelachaud, 2009), musicians resort to subtle changes in sound, such as variations in tempo, pitch, loudness and timbre, thus generating a specific expression to be transmitted (De Poli, 2004; Arfib et al., 2005).

1.3. Methodology and results

1.3.1. Functioning of Arcontinuo

The Arcontinuo has a curved touchable surface that works with Frustrated Total Internal Reflection (FTIR) technology.

> FTIR technology is based on optical total internal reflection within an interactive surface. Electromagnetic waves transmitted within an inner material are completely reflected at its boundary if both the inner material has a higher refractive index than the outer material and the angle of incidence at the boundary between the materials is small enough. Common FTIR set-ups have a transparent acrylic pane with a frame of LEDs around the side injecting infrared light. When the user touches the acrylic, the light escapes and is reflected at the finger's point of contact due to its higher refractive index; an infrared-sensitive camera at the back of the pane can clearly see these reflections (Schöning et al., 2008).

The most updated version of the Arcontinuo has a matrix of 1512 Infrared sensors, instead of a camera, that are read with the help of an FPGAs instead of micro-processors, that allow a high-frequency sampling of the sensor data (R. F. Cádiz & Sylleros, 2017).

In older prototypes, the algorithm to detect fingers' positions was implemented in a software in a computer, that implied sending all the sensor images by UART. In the current version, as it has much more sensors, to send all the data at the reading frame rate implied a big error probability. For this reason, it was necessary to implement the image processing algorithm to detect the fingers' positions directly in the FPGA in order to reduce the information that the device sends to the computer.

1.3.2. Detection of Fingers in Arcontinuo

The first step was the implementation of a signal processing algorithm that allow us to detect the fingers touching the surface from the pre-processed images, as the one shown in

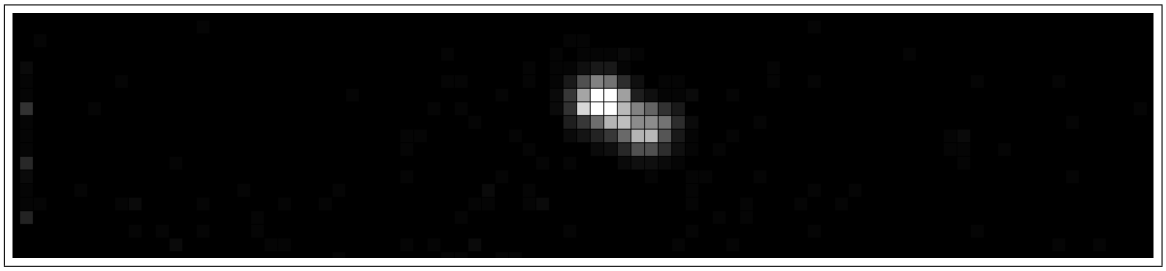
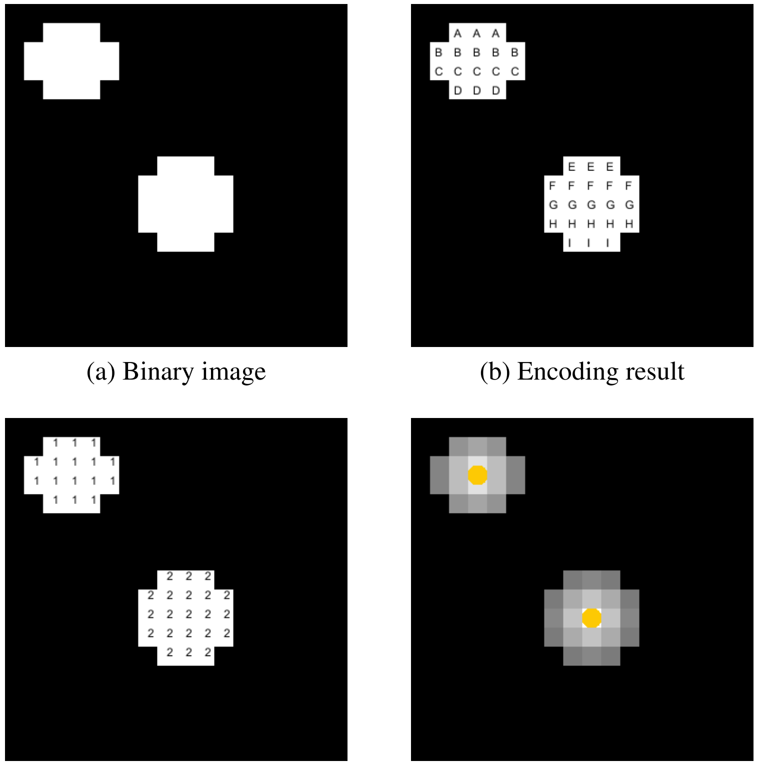


Figure 1.4. Pre-processed image of Arcontinuo

1.4. The pre-processing stage generates an image with the mean values of each sensor in a passive state. At the beginning, when the device is powered-on, the mean value and the standard deviation of each sensor are calculated. Then, in each frame the sensors are read; if the mean plus the standard deviation is exceeded, the pixel take the read value minus the mean and the standard deviation in passive state; in another case, the pixel is set as zero. All the values are saved on as a gray image on RAM, which have the sensors values in the proper order.

To start, the algorithm of figure 1.5, based on Run Length Encoding (RLE) and Components Connected Labelling (CC), was implemented (C. T. Johnston & Bailey, 2008; Bailey, 2011). The gray image is transformed into a binary image, which identifies pixels with fingers. From the binary image, rows are extracted with a run-length encoding algorithm, so we can connect active zones horizontally. Then, we detect if zones of close rows are connected and assign the same label when it is the case. The centroids of the fingers are obtained considering the pressure observed for each active pixel with the same label.

The image processing algorithm begins with the binarization of the gray image considering the Thresholds RAM, which saves a different threshold for each row. The thresholds are different for rows in order to compensate the differences of the distribution of light. The user defines a threshold for the external part of the image and a percentage of decrease at the image's center. In this way, it is possible to define thresholds per each row that depend of the distance to the center and are smaller at the center, where the infrared light is



(a) Binary image (b) Encoding result

(c) Labelling result (d) Centroids result

Figure 1.5. Connected Components Algorithm for FPGA. From the binary image, we extract rows and encode active zones with a run-length encoding algorithm, so we can connect active zones horizontally. Then, we detect if zones of close rows are connected and assign them the same label if it corresponds. The centroids of the fingers are then obtained considering the pressure observed for each active pixel with the same label.

weaker. The Binary RAM saves with a 1 the pixels with a higher value of its threshold and a 0 in another case.

After this, the binary image is read and filters are applied to improve it. To read the RAM in an ordered way, the image's rows are extracted one by one with the module Image-To-Row, that extracts one full row and sends it to the other modules with the index of the row. Morphological filters (dilation and erosion) are applied to close holes in the image that noise can produce.

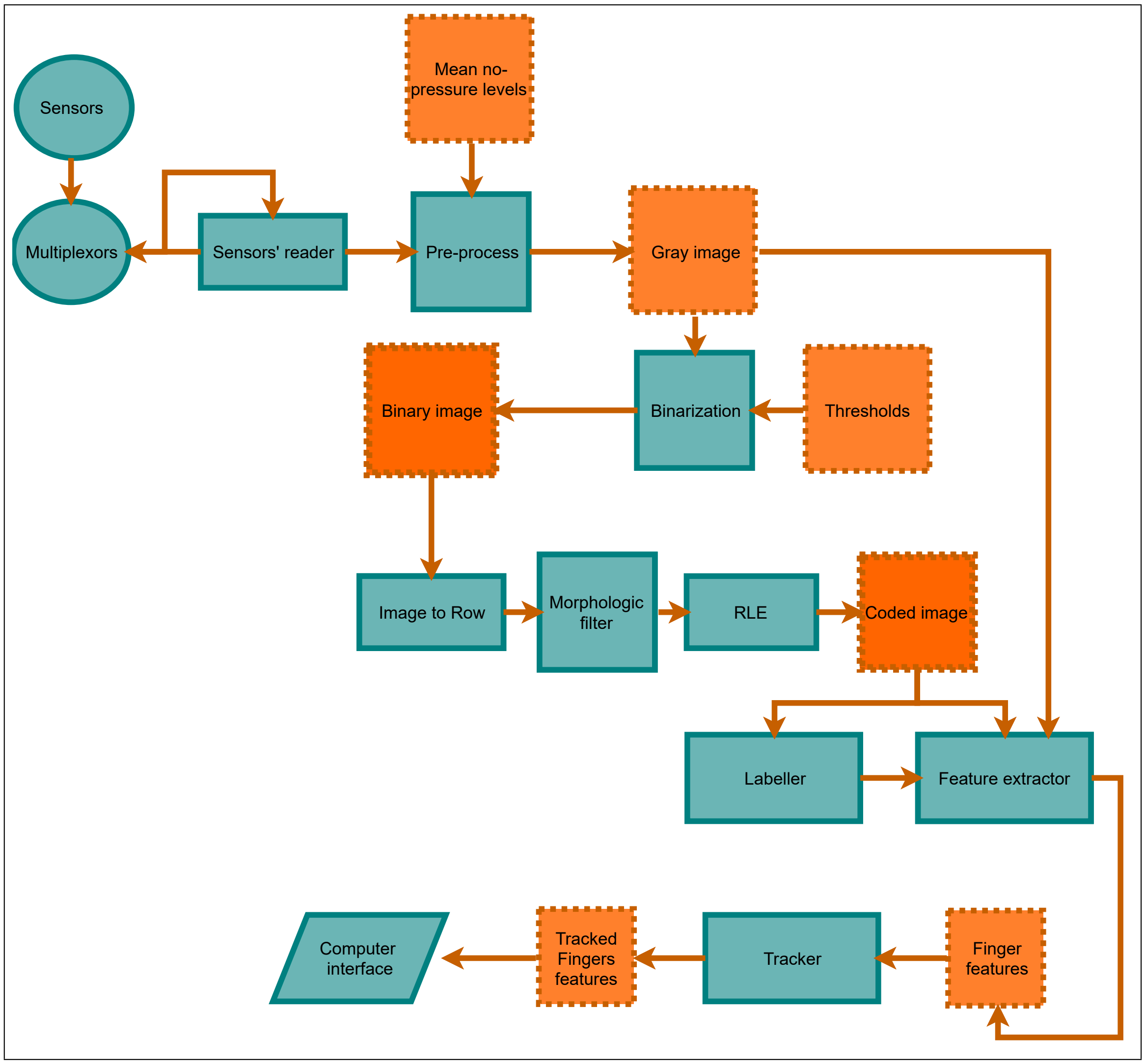


Figure 1.6. Block's diagram of FPGA's circuit for reading sensors and detecting fingers positions. The sensors are read with a set of ADCs and the block *Sensors' reader* sets which sensor are connected. For each sensor, the value is compared with a statistical version of the sensor, that establishes the non-active levels in RAM. *Mean no-pressure levels*. The processed values of sensors are recorded in RAM memory denoted as *Gray image* and compared with the *Thresholds* RAM by the block *Binarization*, resulting the *Binary image*, which saves which pixels have a finger pressing. *Image to Row* reads one full row and gives it to *Morphological filter*, which eliminates holes. Then, *RLE* encode rows to say how many zones with fingers a row has and their positions in the *Coded image*. *Labeller* compares the positions of fingers in correlative rows and give the corresponding labels to the zones, while *Feature extractor* save values of each row in order to calculate the finger's position and saves it in *Finger features*. *Tracker* gives an id to each finger and saves sample by sample the new positions of the fingers in *Tracked Fingers features*. Finally, these values are read and *Computer interface* sends them by UART to the outside world.

In order to tag the different zones of the image, a Run Length Encoding (RLE) algorithm was implemented. Each row is coded with the RLE module, that tells in which column does the zone starts and ends with each finger. In this way, the quantity of information is reduced and now it is just necessary to determine which zones are part of the same group. To do that, the algorithm of Connected Components (CC) is implemented in the Labeller module, which gives a tag to each active part of the image. The CC algorithm identifies when two different parts intercept in order to produce the same tag.

The Feature Extractor module receives the assigned tags from the Labeller and calculates the relevant features from fingers touching the surface, which are shaped as blobs in the image. For each blob, it is important to obtain the centroid, area and pressure. To obtain the features, for each part we calculate the sum of pixel values z_i, the center (cx_i,cy_i) in axis X and Y weighted by z_i. Then, considering each zone i labeled with the label l, we can obtain the centroid in X as:

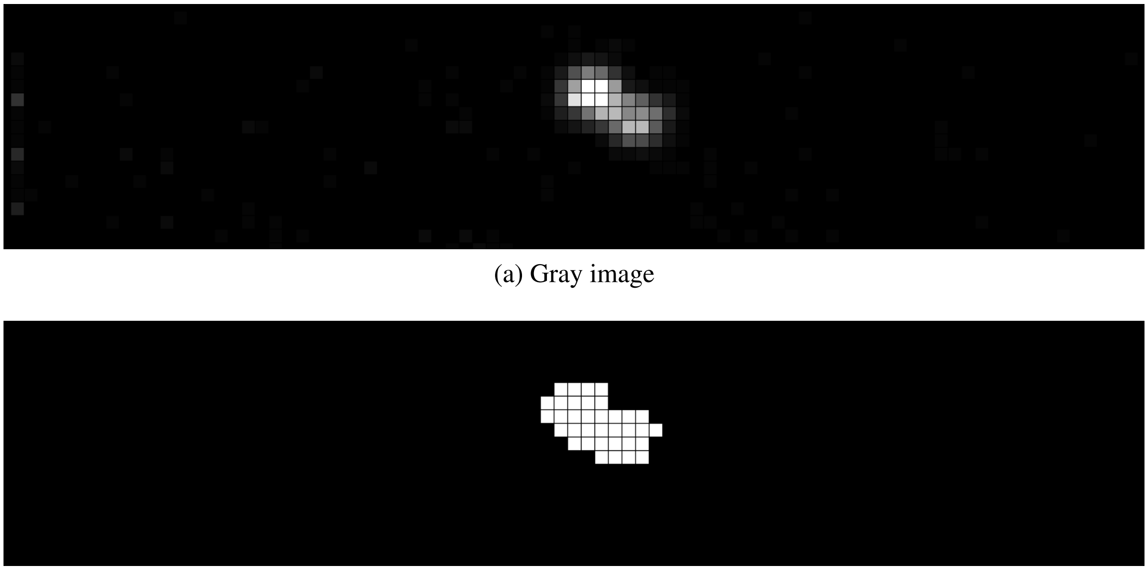
$$x_l = \frac{\displaystyle\sum_{i \in I} cx_i * z_i}{\displaystyle\sum_{i \in I} z_i}$$

And the equivalent process is applied to obtain the center in Y.

As figure 1.7 illustrates, this algorithm can not separate fingers that are too close; to solve this situation, a watershed algorithm was implemented especially to divide near fingers. This algorithm, explained in figure 1.8,

> considers the pixel values of an image as a topographical map and segments an image based on topographical watersheds. [...]A droplet of water falling on each pixel within the image. The droplet will flow downhill until it reaches a basin where it will stop. The image is segmented based on grouping together all of the pixels that flow into the same basin(Bailey, 2011).

The implementation of the watershed algorithm, shown in figure 1.9, considers some changes in the algorithm. For pixels that exceed the threshold, the position of the maximum value in a 3x3 neighborhood is calculated and saved in a RAM. Then, the labelling process, for each pixel has the following possibilities:



(a) Gray image

(b) Binary version

Figure 1.7. Binarization of closed fingers. The connected components algorithm can't distinguish between this two fingers that are so near.

(i) *no label and maximum without label:* a new label is assigned for the evaluated pixel and the maximum

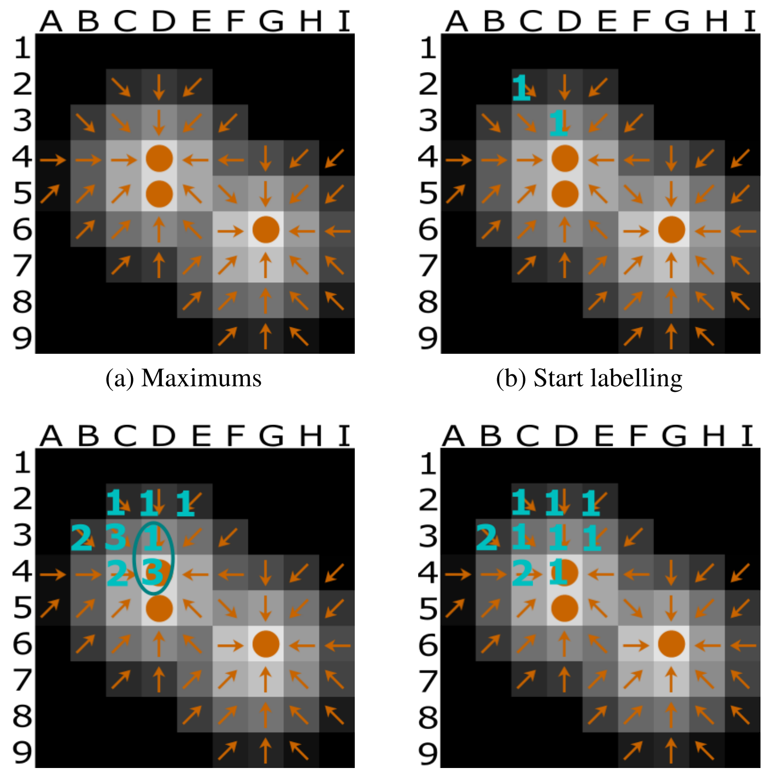
(ii) *no label and maximum with label:* the pixel receives the label of the maximum pixel in its neighborhood

(iii) *label and maximum without label:* the maximum receives the label of the pixel

(iv) *label and maximum with label:* the labels from the pixel and the maximum are merged

To make the merging process, a RAM saves the equivalences of labels. In parallel, the Feature Calculator sums the weighted positions of the pixels with the same label and sends the information of position of fingers when all the image is reviewed.

Finally, it was necessary to implement an algorithm that allow us to track fingers and give them the same id. We implemented the algorithm proposed by Huang et al. (2015) for tracking with a large number of points. The data from the past frame is saved in RAMs



(a) Maximums

(b) Start labelling

(c) Merge condition

(d) Merging results

Figure 1.8. Watershed Algorithm for FPGA. From the gray image, for each pixel bigger than the threshold, the position of the bigger pixel in the neighborhood is determined, as figure 1.8a illustrates. All pixels start without a label, and repeat the label of the maximum closest pixel. In figure 1.8b we started evaluating pixel C2, which should have the same label of D3; as they both didn't have a label, a new label was assigned. Then, D2 and E2 receives the same label as D3, because it is their maximum. In figure 1.8c, when it is time to check pixel D3, which maximum is D4, we find that it has a different label from its maximum, so both labels are merged and label 3 become 1, as figure 1.8d.

and each frame positions from old blobs are compared with the new ones. We constructed a table that reflects the distances between two blobs. We used the Manhattan distance:

$$D_{ij} = |x_i - x_j| + |y_i - y_j|$$

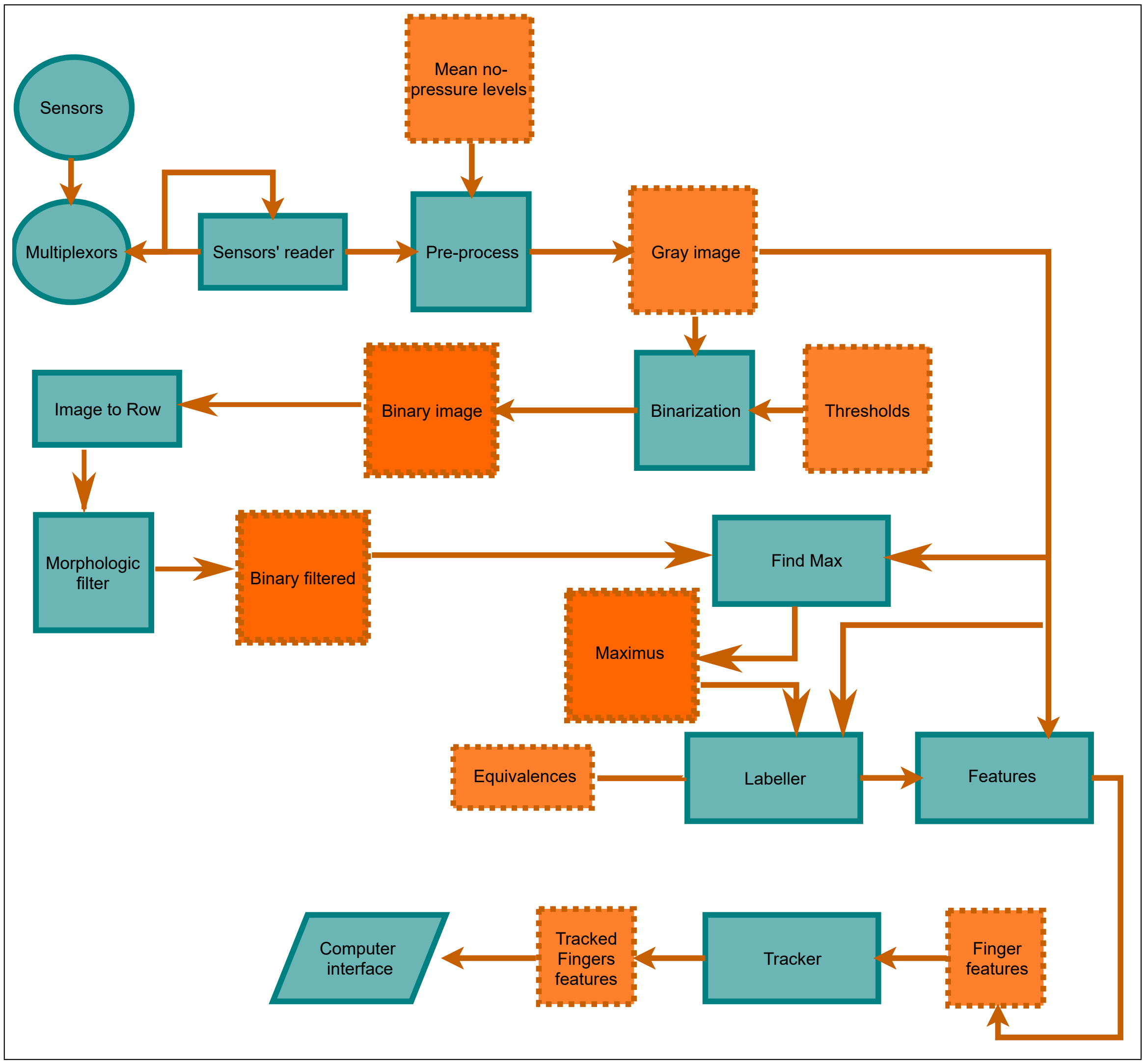


Figure 1.9. Block's diagram of FPGA's circuit for watershed algorithm. In this case, the maximum position in the 3x3 neighborhood is searched and saved in the RAM Maximum, then, the Labeller FSM assign the labels that group following the flow and saves in Equivalences the merging. Features accumulate the weighted positions of pixels with the same label and make the merging process.

This distance measure was chosen instead of euclidean distance because of its easier implementation in FPGAs. As figure 1.10 illustrates, once all distances are calculated, the lowest valid distance is searched. If the minimum distance is lower than a threshold, the ID of the corresponding old blob is assigned to the new blob and all distances associated are discarded. The process is repeated until all old blobs are assigned or all new blobs without an ID have distances that exceed the threshold. If after this process there are still new blobs without ID, they receive a new ID.

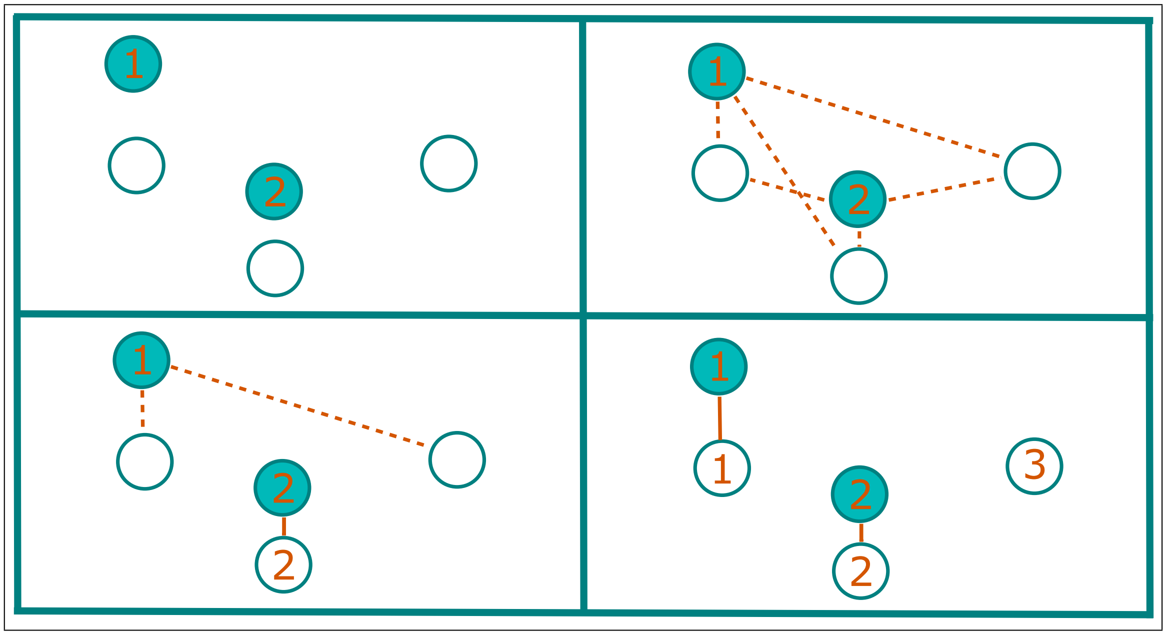


Figure 1.10. Tracking algorithm implemented for Arcontinuo in FPGA. The old blobs (coloreds) are compared with the new ones (whites). The first step is calculating the distances between all old blobs positions and the new ones. Then, the shortest distance is searched for; if it is lower to a threshold, the ID of the corresponding old blob is assigned to the new blob and all distances associated are discarded; in this case the blob 2 is near of the medium white blob. The process is repeated just for distances between the rest of the old and new blobs. In this case, the distance between blob 1 and the left blob is the lower, so it is assigned with and ID of 1. Finally, all old blobs are assigned but the last new blob doesn't have and ID, so it receives a new one.

1.3.3. Analysis of DMIs based on multi-touch

An important part of this study was the analysis of interaction in different DMIs based on touchable surfaces, which is presented in chapter 2. A list of 28 musical instruments were reviewed in order to understand the current strategies that the DMI's designers use when they work with touchable surfaces. Some concepts from Human Computer Interaction (HCI) were use in the analysis.

One important result, presented on section 2.3, is an expanded version of the DMIs' model. The presented version divides the controller stage into interface and processing frameworks, that allow us to analyze and design this two process separately. The first consists on the organization of the playable surface and the determination of what kind of values can be directly obtained from it. Processing frameworks consist on the interpretation of the information obtained from an interface framework. It determines what kind of interactions between multi-touch points and their evolution in space and time will be considered. Both frameworks give information about the musician's possibilities in relation to the interface, and can help us to understand how the musician intends to move in the interaction event. Moreover, interaction and processing frameworks are tools that define how the user will construct his own mental models and how they understand that the mapping and synthesis stages work.

The observed strategies were divided in different levels of interaction and processing and the interactions of them were analyzed. Interaction was divided into keys, sliders and multi-dimensional zones, as three levels of interaction that can be obtained in interfaces with touchable surfaces. These three levels represent different kinds of information and possibilities in the usage of controllers. Processing, by its side, was divided in vertex, polygons and gestures, as different levels of information that can be obtained with algorithm that process the information. Finally, the current usage of the strategies and their possibilities were compared. For each level of processing and interaction there were defined affordances, actions that users can achieve when they interact with the DMI, and constraints, limits that the controller has when someone interacts with the DMI. For example, a key-based strategy affords the action of pressing and has the constrain of selection of one value and processing frameworks create the possibility to change the synthesis in subtle ways, because they provide information about low and high level changes on the musician's movements that can be incorporated into a posterior mapping stage.

We detected that the usage of different kinds of interaction frameworks were not widely connected with their affordances and processing frameworks have not been exploited. The most common use of keys is to select pitch directly, so this is not very different from the direct manipulation of a computer keyboard. On the other side, sliders and multi-dimensional frameworks have commonly been used to control general parameters of the system or features of the sound synthesis. Processing frameworks have not been very popular but its usage is usually correlated with their affordances: tracking has been used to understand beginning an ending points of any touch action, postures and gestures are connected to a multi-dimensional control of timbre and amplitude. Nevertheless, in many cases, frameworks do not appear to be designed to lead movements based on musical metaphors: the gestures and postures are not directly related with sound changes. Moreover, the use of gestures on DMIs are usually related with gestures used on a typical tablet and mobile apps.

In sum, chapter 2 establishes an starting point to propose an strategy for multi-touch DMIs. The incorporation of a processing stage that allows for a better understanding about gestures can expand the possibilities to design DMIs based on how they both work. For example, is clear that a multi-parametric control of sound requires a processing framework that would let us analyze the movement in a multi-dimensional way.

1.3.4. A proposal to play multi-touch surfaces

The second part of this research, presented in chapter 3, focuses in proposing a way to play DMIs based on multi-touch surfaces, such as the Arcontinuo. A key aim was to find a way to incorporate expressiveness in the system.

As a lack of the understanding of expressiveness in multi-touch gestures was detected, it was decided to search for a system that would helps us to analyze it analytically in order to incorporate it to the mapping process. Some DMIs were studied in order to establish which descriptors of gestures were used in them. We realized that, in this case, the most common are low level descriptors of movements, that refer to dynamics and geometry of the fingers' gestures, as distance, acceleration, velocity, angles or time of the movement. In general, multi-touch instruments didn't use a mathematical description that take into account high level understanding of expressiveness.

Other DMIs and interactive systems were taken into account to have an idea about how to incorporate expressiveness. The Laban model of movement was selected in order to analyze fingers movements in a touchable surface, as it has been widely use in musical and artistic contexts. Specifically, effort and shape descriptors were used to analyze how the movement is done, as they allow us to distinguish between several features of a gesture: strong vs light, sustained vs sudden, direct vs indirect or bounded vs free.

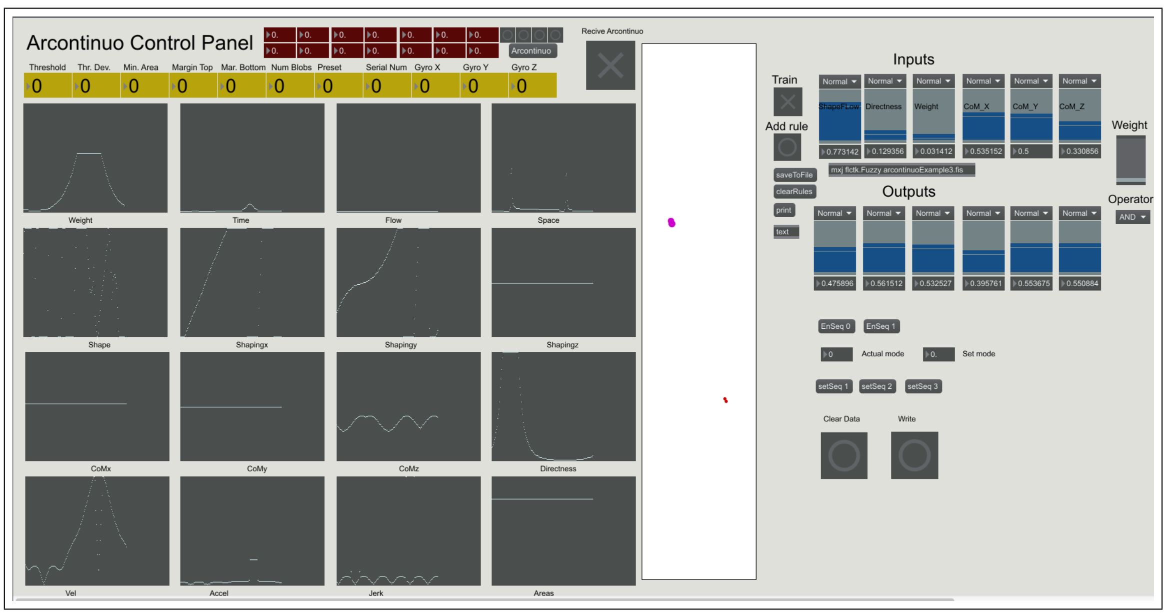


Figure 1.11. Implementation in Max MSP. The system receives the information from the Arcontinuo, in this case by the MIDI protocol, and calculates the descriptors that the user decides to use. The descriptors are the inputs for the fuzzy system, which work following the rules that the user previously defined. These rules can be modified within the patch. The outputs of the fuzzy system may be connected to any synthesis tool.

In this thesis, we present a system, implemented in MaxMSP, that allows for a real-time mapping of expressive gestural analysis into sound synthesis parameters with a fuzzy logic engine. It is a tool that can be used for any user who wish to control any sound synthesis system with any multi-touch system. Users can select what descriptors they will use and set the rules with easy linguistic parameters, without mathematical analysis.

The gestural descriptors were extracted in real-time on MaxMsp receiving the finger's positions from the Arcontinuo and a simulated version of it was also implemented. The descriptors were tested, with the simulation of Arcontinuo and the real one, and we checked that the they worked properly and described appropriate changes in gestures. Figure 1.11 shows the implementation in MaxMsp and the changes that some values experiment. The user can select what gestural descriptors he wants to use as inputs for the mapping process.

To map the motion descriptors to synthesis features, a fuzzy logic system is proposed. The Fuzzy Logic Toolbox (R. Cádiz & Kendall, 2006; R. F. Cádiz & Gonzalez-Inostroza, 2018) was used to implement the system in MaxMSP; as it allows for the implementation of real-time fuzzy systems in computer musical environments. Users can set the rules in the system in real-time or import them from a text file created in MATLAB or similar software.

The system was tested with three different strategies to define rules: based on descriptors, based on a set of gestures and based on musical metaphors. Simple mappings that connect one input with one output were tested satisfactorily, but they do not allow for cross-coupling mappings. Also, it was proven that the system responds for specific gestures that can be related by rules analyzing which descriptors change. Finally, both strategies were mixed in order to define rules related with musical metaphors.

Chapter 3 presents a full proposal for processing and mapping of multi-touch information following an expressive understanding of gestures. This is an significant advance, based on research, for the development of Arcontinuo that opens the possibilities to define a mapping stage that would work in stage for any sound synthesis.

1.4. Conclusions

On one side, the implementation of the algorithms for the detection of fingers in FPGA for the Arcontinuo works satisfactorily. It is possible to detect up to 10 fingers at a 120 fps rate. The system allows to use such a touchable surface in musical contexts. In this regard, this research helped the development of Arcontinuo.

On another side, this research represents some contributions for the design and analysis of DMIs in general. Design of DMIs is not a simple task because it involves many relations between the musician, the DMI, other musicians and the audience. For these reasons, the interaction quality must be a central objective in a DMI design process. A central point in this interaction is the expressiveness that the musician can achieve with the DMI. In order to do that, the expressive analysis of gestures can give an understanding about emotional communication, which can be required to make a sound synthesis related with expressiveness.

In this research, we presented a model that allows for the understanding and designing of interaction events on touchable DMIs. Musicians interact with a specific interface that can be controlled with interaction and processing frameworks, and as such it is important to think and design what happens on these stages and how they relate to posterior mappings and synthesis stages. The model enables to analyze differences of interaction that are not totally clear with the classical model of DMIs.

Sound must be controllable by the musicians, meaning that the interaction and processing frameworks and the mapping stage must give sufficient possibilities for subtle changes. Designers of DMIs have to think what is possible with the specific instrument and what the system is facilitating. In consequence, frameworks should allow to define parameters in a precise and varied way; processing can multiply the possibilities to understand the gestures and use it in the mapping stage.

In order to synthesize digital sounds on stage, is important to develop systems that would let us analyze gestures at low and high order in real-time. This research uses common low level descriptors for finger gestures and adapt full-body movement analysis descriptors. In chapter 3 it is shown that descriptors of effort and shape, based on Laban's theories, are compatible with touchable surfaces and give a good approximation that let us distinguish expressive features of fingers' gestures. An implication of this is the possibility of mappings between expressiveness features of gestures for artistic expression.

Fuzzy logic was shown to be a good option for mapping with gestural information in DMIs. Mapping of gestural information, which usually consider multiple inputs and outputs, require strategies that let musicians modify the system's behavior easily and understand and predict how it works. Approaches like functions setting do not give clear information about it and can confuse users. On the other side, recognition of gestures, based on hidden Markov models or neural networks, usually respond satisfactorily just to pre-trained gestures. Fuzzy logic, in contrast, allows for a mapping based on multi-parametric common-sense rules. Additionally, rules can have different degrees of relevance, that may facilitate the preponderance of some rules over others.

As it was exposed in this thesis, we were able to complete the research goals previously defined. To start, two algorithms to detect fingers were implemented satisfactorily in the Arcontinuo. Then, to generate a system for sound synthesis, 28 DMIs were analyzed and we determined advantages and disadvantages of different current strategies in DMIs with touchable surfaces. These ideas remain in a processing stage that include an expressive analysis of gesture which is used in the mapping stage. Finally, we developed a system that incorporates expressiveness for multi-touch DMIs and, in particular, for the Arcontinuo.

1.5. Future Work

The model presented on chapter 2 may be generalized for DMIs and a similar analysis of interaction and processing frameworks can be done. There is a potential contribution of the proposed model in a general version, as it can explain process that are not observable in the classical model of DMIs, some other features can be designed or analyzed with the presented model.

The descriptors should be tested with others multi-touch surfaces, such as tablets. This would prove the usage of expressive descriptors based on Labanotation in multi-touch environments. Some touchable interfaces won't need adaptations but for others we should consider some changes, especially for the usage of Z and area.

In order to use the presented work in a musical context, it is necessary to find rules that would ensure a pertinent interaction on stage. The design of behavior rules can initially be made based on common sense or theoretical analysis, but we consider that it is primordial the incorporation of users in all the process. A further study with more focus on the construction of rules is therefore suggested. In this case, is important to test the interaction between musicians and the specific DMI analyzed.

Finally, we think of primordial importance the realization of a testing experiment with musicians on stage. In the particular research of the Arcontinuo, a synthesis structure should be defined beforehand, and how it would change should be asked to musicians and potential users of the instrument. Also, it is important to test the interactions between the instrument, the musician, other musicians and the audience.

CHAPTER 2. UNDERSTANDING INTERACTION IN MUSICAL MULTI-TOUCH SURFACES

2.1. Introduction

[1]Digital Musical Instruments (DMIs) are human-computer interfaces which allow musicians to control a computer synthesis process (Wanderley & Depalle, 2004). DMIs have expanded the possibilities for live performance of computer music because they allow the control of real-time interactive sound synthesis. These interfaces have evolved in parallel with other technologies that have improved the interaction between musicians and computers. Today, when we can see touchable surfaces everywhere, many DMIs have incorporated touch technologies in order to capture the movements of hands and fingers.

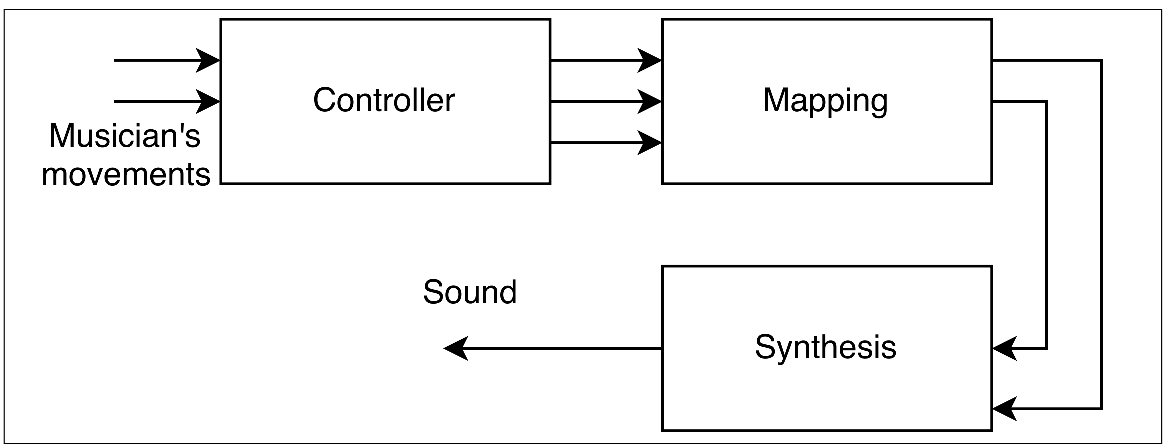


Figure 2.1. Classic model of DMIs incorporates controller, mapping and synthesis stages (Wanderley & Orio, 2002; Wessel & Wright, 2002). The controller stage does not incorporate information about subdivisions, interface controls or signal processing sub-stages that would allow the understanding of what the user is doing.

[1]This chapter is published as González-Inostroza, M., Sylleros, A., Cádiz, R. (2017). Understanding interaction in musical multi-touch surfaces. In 2017 *icmc/emw - 43rd international computer music conference and the 6th international electronic music week*

Typically, DMIs are been understood as a three-stage process as shown in Figure 2.1 (Wanderley & Orio, 2002; Wessel & Wright, 2002). This model divides the process on three stages: input controller, mapping and synthesis. The first one represents the physical interaction and inputs for the system. The second stage translate the musician's movements into musical features or parameters that are the inputs for the synthesis stage, which generates sound. In consequence, this model allow us to divide the design of DMIs on modular and semi-independent processes, but it does not include specific details about the interaction between musician and instrument.

Research in Music and Human Computer Interaction (HCI), known as Music Interaction, has contributed to a deeper understanding of DMIs. HCI provides tools to analyze musicians' activities and to evaluate DMIs as interfaces that control a computational process (Holland et al., 2013). We think that the usage of some theories taken from the field of HCI would improve the analysis of the interaction process on DMIs, as we discuss now.

In HCI, *mental models* are the user psychological explanations about how a system works, that helps enlighten what the system will do in response to an specific action. They can be constructed by *metaphors* and modified by usage. Metaphors are representations of a system that define rules of its behavior. They reference a well-known system in order to facilitate the understanding of a new system performance (Wickens et al., 1998). On the other hand, HCI also considers *frameworks*, which are conceptual or physical systems that structure a process or a system with the aim of improving the interaction (Mooney, 2010).

HCI designers must facilitate the relationship between what the user wants and the action they must produce in order to obtain what is desired. In consequence, they must help building correct mental models that would predict what the system will do in response to an specific action. As we indicated earlier, frameworks must be applied to the design for a conceptual understanding of the system and they are based on metaphors that help in making the system behavior transparent (Wickens et al., 1998).

As it was proposed in (Jordá et al., 2010), touchable surfaces bring actions from the physical world to a virtual representation, meaning that the usage of real-world metaphors from the users' movements and actions can provide easier ways to use the interface. However, in the case of DMIs, metaphors have been in most cases centered on mapping and synthesis, because they are considered the core of the instrument (Magnusson, 2010). We argue that a more careful DMI analysis and design of the interface can give a better understanding of the system and incorporate the users' movements in a more natural way.

Interfaces use physical and conceptual frameworks that define the interaction and possible actions, which are tools themselves. Musical frameworks have been used on compositional and performing contexts and influence the way the music is done. A violin, a synthesizer or a digital audio workstation software are examples of physical frameworks. Also, conceptual frameworks are common in music, for example a twelve-bar blues or a metric rhythm (Mooney, 2010).

Any framework has *affordances*, actions that the users perceive they could do with the system and are possible to achieve (Norman, 2013). Also, they have *constraints* that avoid making other actions. Affordances and constraints are studied in order to understand what a user could do in the presence of the system (chen You & Chen, 2007). Mooney (Mooney, 2010) takes the violin as a good example: "Since the violin comprises strings stretched between two fixed points, so it is acoustically predisposed to produce pitched sounds. Accordingly, if we were to look at the violin repertoire, across all the genres of music and styles of performance we would probably find, on average and generally speaking, that it is used to produce pitched sounds more often than noisy sounds".

In this paper, we propose an expanded model of DMI that would work for touchable DMIs in order to understand and classify different processes that are present in these kinds of interfaces. Section 2.2 provides a discussion of interaction quality for the case of DMIs. In section 2.3 we detail this expanded model for the understanding of interaction on touchable DMIs. Section 2.4 provides a level-based classification of interaction and processing frameworks and we study how controls and processing influence the musician-instrument interaction. In section 2.5 we analyzed frameworks used by twenty eight well known multi-touch DMIs and discuss how different levels of interaction and processing should be applied on touchable DMIs. Finally, we present a general discussion and the conclusions of our work.

2.2. Interaction on DMIs

Interaction quality can be understood as an event "where the subject, driven by personal meaning, is concerned with the material culture (context) to do something satisfactorily when dealing with objects" (Sylleros et al., 2017). Users perceive value on interaction events that help to understand when an object works better for them in an specific context. Sylleros *et. al* classify the value in three categories: operable-functional, visceral-sensory and reflexive-symbolic. The first value relates the object's features with advantage when the user manipulates and operates with the object. The second value refers to sensory dimensions of the object that affect the subject's perception about its behavior. The last value refers to subject understanding and explanations about the object operation. All these values are connected and their relations affect perception of quality of the interaction. Therefore, the inclusion of potential users of the DMI in the design and validation process is crucial to obtain better designs (Sylleros et al., 2017).

In the context of the operable-functional value, it is desirable that DMIs allow for expressiveness and the development of an individual style. Expressiveness has been described as subtle variations in tempo, timbre and pitch that help the musician to communicate feelings and intentions (Arfib et al., 2005). In consequence, a DMI must help controlling the sound even if it only implies tiny changes in the sonic output.

DMIs require some basic common features such as ergonomics, precision and low latency. The instrument must be comfortable in terms of its structure, size, weight, construction and the movements that the user has to do to operate it (Sylleros et al., 2014).

In this case, operable-functional value when combined with visceral-sensory value affect the perception of quality. Also, the instrument has to ensure a minimum level of accuracy, which means that the detection of the movements must be precise in space with small amounts of computation time.

Who plays the DMI, other musicians on stage and spectators interact with the musical instrument and give reflexive-symbolic value to it. Therefore, the musician's movements must allow the understanding of what he/she is doing and how he/she is manipulating the sound. This is where metaphors help musicians and audiences to understand how a DMI is working. Metaphors related with known musical movements will allow for better mental models about what sounds will be triggered by any specific movement.

In order to make better DMIs based on touchable surfaces, we have to pay special attention and carefully design how the musician will interact with the interface. Following some ideas we can borrow from HCI practice, we propose to build DMI interfaces that ensure affordances for musical purposes. To achieve that, we now study interaction in the context of touchable DMIs. It will help us to design touchable DMIs centered on interaction quality, because we can focus on value, specifically at the operable-functional and reflexive-symbolic levels.

2.3. An expanded model for interaction on touchable DMIs

We propose a model aimed to expand the understanding of touchable DMIs. The classic DMI model, shown in figure 2.1, makes explicit what happens once the user's movement is finished, but it does not provide means for analyzing or designing the interaction between the musician and the interface. How interaction is analyzed depends on the interface, which means we must find common characteristics for multi-touch surfaces used for DMIs.

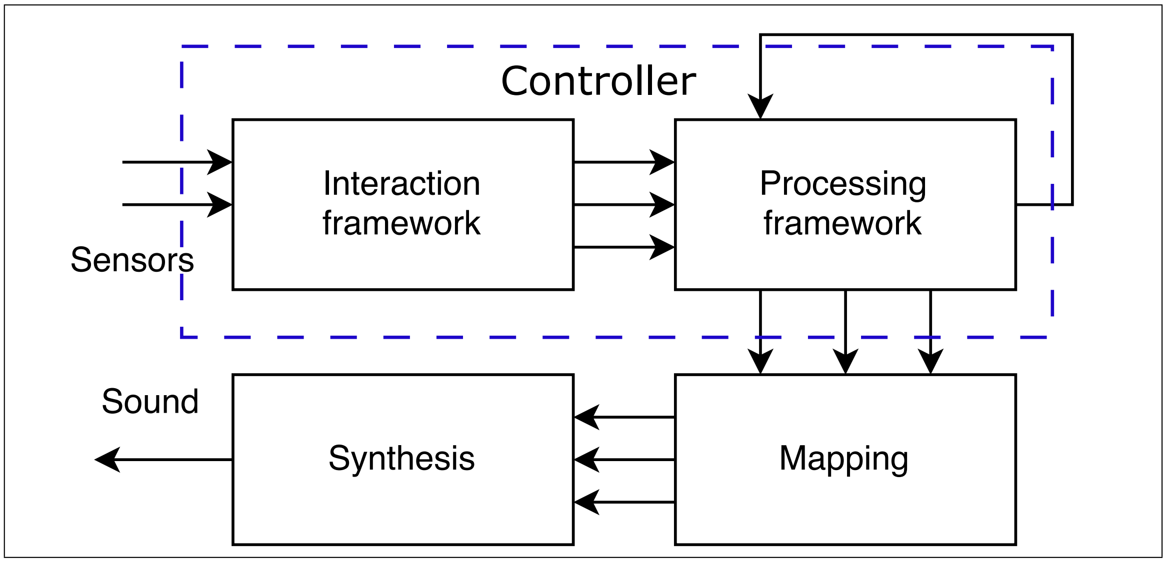


Figure 2.2. Our scheme aggregates an interaction and processing framework to the classic model of figure 2.1. The interaction framework refers to controls that are present in the system and the processing frameworks interpret received information for a better understanding of the user's movements.

We propose the addition of two levels of frameworks to the classic DMI model, as figure 2.2 displays. DMIs interaction designers should consider, in addition to the classic model, interaction frameworks and processing frameworks. This modification divides the Controller stage of figure 2.1 into these two frameworks that represent different levels of interaction with the interface. These frameworks define how a musician will interact with the system, what movements will originate changes on the produced sound and what won't change the result.

Interaction frameworks consist on the organization of the playable surface and the determination of what kind of values can be directly obtained from it. Sensors give information about the finger's position and, in some cases, the pressure being applied to the surface and they can be organized in zones with specific meanings in the system's operation. These zones are controls of the system and help the user to understand which parameters are being obtained from them. This stage can be defined on a physical layer, for example some DMIs have keys on their structure, but they can exist on a virtual space, as grids on tablets.

Processing frameworks consist on the interpretation of the information obtained from an interface framework. It determines what kind of interactions between multi-touch points and their evolution in space and time will be considered. Some approaches incorporate the use of machine learning techniques in order to understand what the musician is doing (Caramiaux & Tanaka, 2013). These dynamic analysis are usually achieved by feedback, as figure 2.2 shows, in order to compare past and present states.

Both frameworks give information about the musician's possibilities in relation to the interface, and can help us to understand how the musician intends to move in the interaction event. Moreover, interaction and processing frameworks are tools that define how the user will construct his own mental models and how they understand the mapping and synthesis stages work.

2.4. Levels of interaction and processing on touchable surfaces

A framework can be studied "as a collection of independent, smaller, frameworks; and likewise, any collection of frameworks can also be regarded as a single, larger, framework" (Mooney, 2010). We have detected that many touchable DMIs, as shown in table 2.1, use frameworks constituted by different kinds of controls or processing strategies that are frameworks by themselves and can be studied as single elements with their own affordances. In order to facilitate the analysis, we propose to classify different frameworks for interaction and processing as levels about what interactions and relations they afford. Nevertheless, we must consider how the smaller frameworks are related to the interaction and processing frameworks in order to establish how the user will interact with the full system.

2.4.1. Levels of interaction

Interaction frameworks can be classified according to the information that can obtained from them and the interaction they afford. We propose a classification of interaction frameworks based on three basic levels: keys, sliders and multidimensional zones. Each level adds new possibilities and expands the number of involved dimensions.

(i) **Keys:** The discretization of the surface by means of keys, zones or buttons that allow to select a singular parameter or (de)active a property. The user can touch any point of the key and the result will be the same.

(ii) **Sliders:** With sliders or knobs the user can change one parameter, but usually the change happens continuously. It is not possible to define an specific value without passing through all the possible values between the actual and the desire value.

(iii) **Multi-dimensional zones:** In this case, the surface is conceived as a multidimensional blank canvas, where it is possible to use continuous or discrete movements. Users can touch a point and then change their position to any other point on the zone or make a continuous change. Absolute or relative positions are used to define parameters, directly by positioning in each dimension. As such, we can obtain as many parameters as dimensions the interface contains, that are typically two or three but it is even possible to define a 1D zone, for example imitating a string.

2.4.2. Levels of processing

We propose also a classification of processing frameworks on four basic types: vertex, polygons and gestures. The first two are static processing of relations of detected points, meanwhile gestures refers to understanding the dynamics of musician's movements in time.

(i) **Vertex:** The analysis of the position of touched points on the surface and their distances or angles. With this approximation, it is useful to set an anchor point that the user will use as a reference. The anchor point could be set as a specific position on the surface, the position of a finger at the beginning of a movement or one finger that can change its position, as described in (B. Johnston et al., 2012).

(ii) **Polygons:** The analysis of touched points on the surface. Absolute and relative positions and angles can be analyze in order to estimate what posture the user is doing in the surface. It could refer to a complex configuration of points, so it may be necessary to rely on some kind of recognition system.

(iii) **Gestures:** A gesture refers to a movement composed by a set of postures over time. It provides high-level information about the musician's movements to understand what is the specific action a musician is doing considering the recent history of observed points. With signal processing algorithms, it is possible to track the position in time of a specific finger, that we propose to label as one-dimensional gestures (1G), or changes on polygons formed by a set of fingers, that implies a multi-dimensional gesture (MG) because the relations between fingers are analyzed with the help of a gesture recognizer algorithm. Using this approximation, it is possible to extract features like velocity, direction, changes of size or even recognize a complex full movement.

2.4.3. Analysis of interaction and processing frameworks

A DMI can combine different kinds of interaction and processing frameworks to build its own framework. When designed this way, it gives flexibility and more options for the musician's movements. However, some kinds of frameworks could be useful for some actions while others are better fitted to control other options, in other words, each framework has his own affordances that help the usage of a specific action to make changes in the system. Which framework is more appropriate depends on the interaction, therefore it changes for each subject, object and context.

Interface frameworks and the specific layout determine what zones are active and how values change. Keys will give us just an specific value, so it is impossible to choose continuous values, but it makes easy to control a single parameter. On the other hand, sliders and multi-dimensional zones afford the selection of continuous parameters, but most sliders can't be changed to an specific value, the change must be continuous. Moreover, multi-dimensional zones give a vector of continuous values with the representation of the position of a touch point and can be changed continuously or discontinuously.

Each interaction framework being analyzed has its own intended action that represents how the user will probably interact with the controls. This inteaction depends on the affordances defined for the framework. Keys can be pressed, so the user will understand he must press and then a single value will be obtained by the system. On the other hand, sliders are draggable, so a continuous movement in one single direction is allowed; vertical or horizontal continuous movements are afforded but it could be a circular movement in the case of a knob. Finally, multi-dimensional zones could be touched with continuous or discrete movements, so the user could press or drag in any direction.

The processing framework, meanwhile, can give more information about relations of touch points in space and time. Vectors analysis gives us positions and angles relative to an anchor point. Polygonal analysis can estimate the geometry formed by touch points. One-dimensional gestures can tell us about velocity, acceleration and other changes suffered by a touch point in time. Finally, gestures analysis could help us with the understanding of the trajectory and changes in geometry experienced by touch points.

Processing frameworks could define relations on the user's movements. Using an anchor point influences movements around it. One-dimensional gestures will bring significance to continuous movements. Polygons and gestures analysis will facilitate that users effectively utilize the actions that the system was designed for because they are meaningful for the synthesis process.

Different types of value requirements will demand different frameworks. Key-based-frameworks could help the user to select an specific value easily, because visual feedback is very clear (Mcglynn et al., 2012). In consequence, controlling pitch exclusively by means of keys could be easy and boring for a musician, moreover, it wouldn't contribute for an expressive performance. Nevertheless, keys could help to easily activate an effect or to define specific parameters. On the other side, sliders would help making a continuous change of a specific parameter, such as the general volume of the system or an effect's input variable, but a multi-parametric change would be very difficult to achieve. With multi-dimensional zones it is possible to obtain many values to control many parameters, to change the timbre or a combination of envelope and pitch for example, but to achieve accuracy on an specific position is complex.

Finally, gestures and its characteristics (as velocity or geometry) could activate and modify some timbral features. The use of processing frameworks create the possibility to change the synthesis in subtle ways, because they provide information about low and high level changes on the musician's movements that can be incorporated into a posterior mapping stage. Nevertheless, DMI designers must reserve some space for communicative movements that don't produce sound, that have been proven to be a important aspect for personal expressiveness.

2.5. Review of interaction and processing frameworks used in touchable DMIs

We analyzed frameworks used by twenty-eight multi-touch DMIs available on the market or reported in the literature in order to understand how different framework elements have been used on their design. Elements present on each framework were classified following section 2.4 and the results are summarized in table 2.1 in chronological order.

In general, we observe that interaction frameworks are not based on their affordances, as described in section 2.4.3. The most common use of keys is to select pitch directly, probably based on the idea of a piano or a keyboard, so is not very different from the

Table 2.1. Levels of interaction and processing frameworks used on analyzed multi-touch DMIs ordered from the oldest to the newest. Interaction frameworks are classified as based on keys (K), sliders (S) and multi-dimensional zones (MD). Processing frameworks are classified as relation of points (R), postures (P), one-dimensional gestures (1G) and multi-dimensional gestures (MG). DMIs can combine any kind of framework to build their own framework.

	Interaction			Processing			
	K	S	MD	V	P	1G	MG
Continuum (Haken et al., 1998)	✓	✓	✓				
Lemur (JazzMutant, 2017)	✓	✓	✓	✓			
PreSenseII (Rekimoto & Schwesig, 2006)	✓		✓			✓	
SoundRose (Crevoisier et al., 2006)	✓		✓			✓	
SurfaceEditor (Kellum & Crevoisier, 2009)	✓	✓	✓			✓	✓
Arcontinuo (R. F. Cádiz & Sylleros, 2017)			✓			✓	
Momu (Bryan et al., 2010)			✓			✓	
Kitara (Misa Digital, n.d.)	✓		✓	✓	✓		✓
Mugician (Roberts et al., 2014)	✓					✓	
Maggic Fiddle (Wang, 2011)	✓	✓	✓			✓	
MT Musical Keyboard (Mcpherson & Kim, 2011)	✓	✓		✓		✓	✓
Soundplane (Madronalabs, n.d.)	✓						✓
ChoirMob (Alessandro et al., 2012)			✓			✓	
TC-11 (Schlei, 2012)	✓		✓	✓	✓	✓	
touch-enabled interface (Ren et al., 2012)			✓			✓	
Linnstrument (Linn & Roger Linn Design, n.d.)	✓		✓	✓	✓		
TouchOsc (Hexler, n.d.)	✓	✓	✓				
Dualo (Dualo, n.d.)	✓	✓				✓	
Capacitance based DMI (Walbeck et al., 2013)			✓				
Roli (Roli, 2018)	✓	✓	✓	✓		✓	
Gibber (Roberts et al., 2014)	✓	✓	✓				
MiniAudicle (Salazar, 2014)	✓						
Orphion (Trump & Bullock, 2014)	✓		✓	✓			
SkipStep (Sarwate & Snyder, 2014)	✓						
TouchNoise (Berndt et al., 2014)	✓	✓	✓				
Touchpoint (Suda & Vallis, 2014)	✓	✓	✓			✓	
Artiphon (Artiphon, n.d.)	✓	✓	✓			✓	✓
Elite (Touch Innovations, 2017)	✓	✓			✓		

direct manipulation of a computer keyboard. To avoid that, some DMIs, like the Orphion (Trump & Bullock, 2014), have used keys as multi-dimensional zones at the same time:

keys select a central pitch that is varied with the distance to the center of the key. Another approaches are the use of keys to (de)activate the system or to select some options. On the other side, sliders and multi-dimensional frameworks have commonly been used to control general parameters of the system or features of the sound synthesis.

Although processing frameworks haven't been very popular, its usage is usually correlated with the recommendations presented in section 2.4.3. One-dimensional gesture analysis, also called tracking, has been used to understand beginning an ending points of any touch action, because it brings the possibility to generate one sound that begins and ends with touch. The detection of relations, postures and gestures is connected to a multi-dimensional control of timbre and amplitude. Nevertheless, in many cases, frameworks don't appear to be designed to lead movements based on musical metaphors: the gestures and postures aren't directly related with sound changes. Moreover, the use of gestures on DMIs are usually related with gestures used on a typical tablet and mobile apps. The exception are interfaces that incorporate elements from acoustical instruments, such as Kitara (Misa Digital, n.d.), Artiphon (Artiphon, n.d.) and the Roli Seaboard (Roli, 2018), that are performed in a similar fashion to the instrument that inspired them.

As was proposed by McGlynn (Mcglynn et al., 2012), DMIs designers have been very conservative in the use of multi-touch surfaces. As we can see in table 2.1, some options are very common and others are used by just some few musical instruments, and there is not a clear evolution through the years when compared to advances in other areas of HCI. Even when we would expect that evolution on computational systems and signal processing algorithms could give us more tools to design DMIs, researchers and developers haven't progressed at the same rate when incorporating interaction and processing strategies that would ensure the interaction quality.

2.6. Discussion and conclusions

Interaction design of DMIs is not a simple task because it involves many relations between the musician, the DMI, other musicians and the audience. For these reasons, the interaction quality must be a central objective on a DMI design process. Incorporating the musicians opinions and experience from the beginning of the design process is very important, as proposed in (Sylleros et al., 2014).

The model presented on section 2.3 allow us understanding and designing interaction events on touchable DMIs. Musicians will interact with an specific interface that can be controlled with interaction and processing frameworks, and as such it is important to think and design what happens on these stages and how they relate to posterior mapping and synthesis stages.

Sound must be controllable by the musician, meaning that interaction and processing frameworks and the mapping stage must give sufficient possibilities for change. Designers of DMIs have to think what is possible with the specific instrument and what the system is facilitating, so it is important to choose frameworks that would allow to define parameters in a precise and varied way.

CHAPTER 3. EXPRESSIVE GESTURAL CONTROL OF SOUND SYNTHESIS IN MULTI-TOUCH DIGITAL MUSICAL INSTRUMENTS BASED ON FUZZY LOGIC

3.1. Introduction

[1]Gestures, from a control perspective, caGestures, from a control perspective, can be defined as observed movements of the body that contain information. All of them have a spatio-temporal component and an intention from the performer in an specific context, that should be understood if we want to use them properly (Jensenius et al., 2010). Therefore, the analysis of gestures in computational contexts does not only require information about spatial position and its evolution in time (Cadoz & Wanderley, 2000) but also contextual information that allows the interpretation of meaning. These different sources of information should be captured by sensors and processed and understood by an algorithm (Schumacher et al., 2016; Camurri et al., 2005).

In musical contexts, gestures have had a central role along history. In the case of traditional musical instruments, the musician's gestures physically generate and modify the sound, allowing musicians to control not only the pitch and loudness, but also very subtle changes in timbre. Such gestures must be mastered by musicians in order to obtain the desired sounds from their instruments. In addition, there are gestures that performers use with the intention of communicating with the audience and other musicians or as a response to the sound (Dahl et al., 2010). In this article, we are focused on the first case: gestures that generate and modify sounds.

Furthermore, in artistic contexts, *expressiveness* has been described as emotional communication (Arfib et al., 2005; Camurri et al., 2001, 2005; De Poli, 2004). In the particular case of music, some authors consider that expressiveness focuses on what message the composer wants to transmit, a message that can be modified by the performer with changes in tempo, timbre and dynamics (De Poli, 2004). From this point of view, gestures must

[1]This chapter was submitted to the *Journal of New Music Research* for peer review

facilitate expression, giving performers the ability to control sound features, producing an expressive sound (Arfib et al., 2005).

In the case of digital musical instruments (DMIs), gestures are usually captured by different kinds of sensors in order to control the sound output (Wanderley & Depalle, 2004). DMIs typically use digital systems to synthesize sounds, and they are able to produce sounds that are not possible to achieve in nature. By means of digital processing techniques, designers of DMIs can connect any movement to any sound, depending of the system features. In consequence, any source of information can be used to generate sound, but the question of how to choose which information to connect to a particular synthesis method does not have a simple answer.

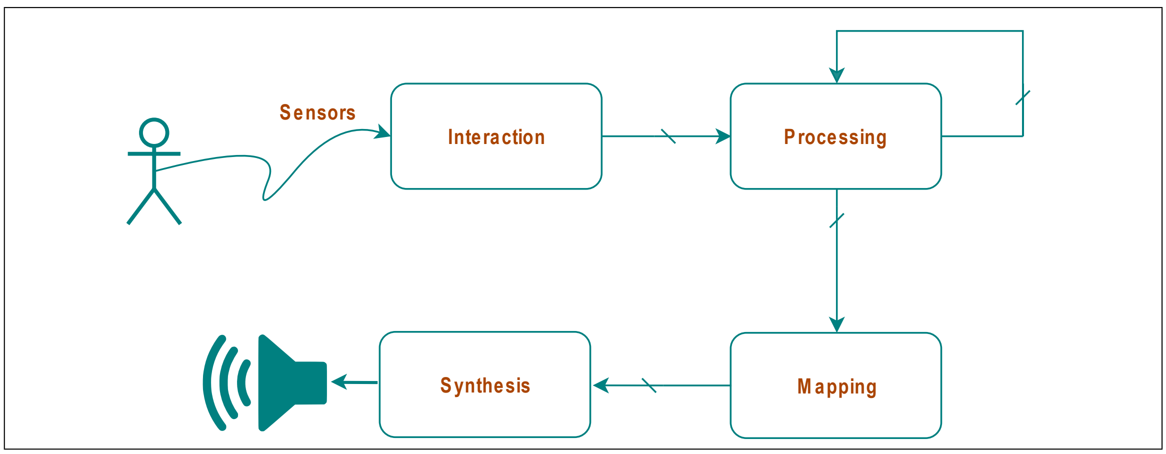


Figure 3.1. The interaction stage refers to controls that are present in the system and the processing stage interprets received information for a better understanding of the user's movements. The mapping stage receives information from both the interaction and processing stages and determines the synthesis parameters. The synthesis stage produces the sound.(Gonzalez-Inostroza et al., 2017)

DMIs have been usually modeled in the literature as a three stage process: controller, mapping and synthesis (Wessel & Wright, 2002; Wanderley & Orio, 2002). Controllers have sensors which capture information from the users' movements. The mapping stage translates this information into synthesis parameters. Finally, in the synthesis stage the sound is produced depending on the results of the previous mapping stage. This model, although it captures the overall process of DMI design, fails to distinguish deeper levels of information about the musician's gestures. In consequence, the authors have proposed an expanded version of this model (Gonzalez-Inostroza et al., 2017), as shown in figure 3.1. This expanded model divides the controller stage into an interaction and a processing stage. The interaction stage establishes the organization and kind of controllers of the interface and the processing stage extracts high level information about the musicians' movements.

DMIs capture basic information from gestures in the interaction stage, usually as spatial positions, but this information can be processed and interpreted as gestures in the processing stage. Usually sensors are chosen and located following some design principles that help us to capture movements of the musician that will be later interpreted as gestures. The importance of the processing stage is that it makes possible the processing of the sensors information in order to obtain more details about the specific gestures performed by musicians.

In order to generate more expressive DMIs, some authors have proposed that connecting not any gesture feature to the sound synthesis algorithm, but including those that are expressive, is essential (Camurri et al., 2001; Maes et al., 2010; Fabiani et al., 2012). In order to achieve that, it is necessary to analyze gestures in terms of their expressiveness, pushing designers to consider descriptors of movement closely related with expressiveness. Also, it is usually necessary to implement mapping stages that work with multiple inputs and multiple outputs. Some approaches have used adapted functions (Maes et al., 2010; Perrotin & D'Alessandro, 2013), rule-based systems, gestures recognition algorithm (Caramiaux et al., 2014; Paradiso, 2014; Geiger, 2006; Street et al., 2016) or fuzzy logic systems (Kreković & Posćić, 2015; R. F. Cádiz & Gonzalez-Inostroza, 2018).

We focus our work in the subset of multi-touch DMIs, which present specific challenges for the incorporation of expressiveness. Current approaches of expressiveness in touchable surfaces typically consider low level features of gestures and one-to-one functions in the mapping stage. We propose the adaptation of expressive descriptors of movement, of both low and high level, for multi-touch applications. We believe that the extraction of meaningful descriptors of finger movements and their posterior analysis in the processing stage can help us achieve a more expressive control in the sound synthesis layer. We also propose a fuzzy logic inference system as the main mapping control strategy. Fuzzy logic can be considered a machine learning technique which allows for non-linear mappings based on expert human knowledge.

In section 3.2, we analyze the common trends in the usage of gestures for DMIs and how they are usually mapped into sound. Section 3.2.2 focuses in the specific case of DMIs that are based on multi-touch interfaces and we propose a way to describe the finger movements in section 3.4.1. A review about strategies for gestural mapping is presented in section 3.3. Our mapping system, based on fuzzy logic, is described in section 3.4 and tested in a real DMI, the Arcontinuo, in section 3.5. Finally, in section 3.6 we discuss the benefits of our system compared to other approaches and some implications for future research.

3.2. Gestures and expressiveness

Expressiveness has been typically understood as the ability to transmit or communicate an emotion (Arfib et al., 2005). In artistic contexts, “behaviors encode content information (the ‘What’ is communicating) and expressive information (the ‘How’ it is communicating)” (Pelachaud, 2009). In the case of music, expressiveness can be analyzed from three perspectives: “the composer’s message, the expressive intentions of the performer, and the listener’s perceptual experience” (De Poli, 2004). We have chosen to focus on the second case, the performer, who modifies the invoked emotion as she performs, by producing subtle changes in sound, such as variations in tempo, pitch, loudness and timbre, thus generating a specific expression to be transmitted (De Poli, 2004; Arfib et al., 2005).

In acoustic music, expressiveness is usually related not just to sounds, but perhaps more importantly to the coupling of the physical gestures with the resulting sonic output (Leman et al., 2005). Musical gestures allow for expressiveness, as they let the performer to finely control the sound in order to generate patterns that work together in a specific context and to transmit a specific idea. Musicians use gestures to continuously control the evolution of the sound, letting them to have control over the expression of the sound (Wessel & Wright, 2002). In addition, gestures themselves constitute an important way to communicate with the audience and other musicians (Leman et al., 2005; Arfib et al., 2005; Paine, 2009).

In the case of DMIs, a careful design must incorporate a gesture analysis task, usually done at the processing and mapping stages. For the processing stage, DMI's designers have tried to understand gestures with different movement analysis algorithms (Paine, 2013; Fabiani et al., 2012; Mancini et al., 2010; Camurri et al., 2005; Loke et al., 2007; Gillian & Paradiso, 2012; Maes et al., 2010; Volpe, 2003; Fenza et al., 2005; Visi et al., 2014; Schumacher et al., 2016). Even though the mapping of gestural information can be designed in multiple ways, there is evidence that a complex mapping strategy, with cross-coupled relations between many input and many outputs, achieves a higher degree of expressiveness, as it would allow users to control multiple variables of sound at the same time and think about the movement they should do instead of how the synthesis work (Hunt et al., 2003).

In consequence, a key aspect of instrument design is the ability to understand and analyze gestures in order to achieve an expressive DMI. Embodied music cognition has shown the importance of gesture in the understanding of musical expressiveness; as planning, execution and perception of an action are similarly represented in the brain, the gesture that generates a sound and the sound itself are understood as the same thing (Maes et al., 2014; Antle et al., 2009; Bakker et al., 2012; Leman & Maes, 2014). Consequently, the expression that musicians achieve as sound can also be found in the gestural process (Maes et al., 2010). It is desirable, then, that DMIs could work as multi-modal interfaces that detect expression in gestures in order to produce expressive sounds (Camurri et al., 2005).

Camurri et al. introduced the term *expressive gesture*, to specifically denote movements that contain information related with emotions; gestures that encompass a particular expression which could be analyzed and later used to control a synthesis process (2001). The problem, therefore, is finding means to analyze the expressiveness of a gesture in a computational setting. Current sensing and processing technologies allow for the development of interfaces capable of interconnecting different levels of movement and expressiveness analysis with the sound synthesis process (Camurri et al., 2005). Moreover, "modeling of expressiveness in gestures, requires proper techniques that capture the subtle temporal/spatial characteristics of expressiveness" (Camurri et al., 2001).

With this in mind, researchers have proposed strategies for mapping expressive features of gestures into synthesis parameters. These features are usually calculated in the processing stage of the DMI, as it was shown in figure 3.1. Some basic approaches focus on detecting the quantity of movement in different parts of the body. Other systems have used particular models of expressiveness. In section 3.2.1 we discuss general implementations of expressive gestures, and the particular case of expressive gestures for touchable surfaces is discussed in section 3.2.2.

3.2.1. Expressive analysis of gestures in DMIs

The expressive analysis of gestures should give us parameters that can make sense in a computational system, in order to use these results in the later mapping stage of a DMI. This expression must be connected to the temporal and spatial features of movements without an explicit meaning (Camurri et al., 2001). These descriptors should be usually calculated in real-time and with low latency.

In the literature we can find some implementations of expressive gestures based on low level analysis of movements. The velocity, for example, is used as a basic and fast descriptor (Paine, 2013; Gillian & Paradiso, 2012). The quantity of motion (Camurri et al., 2005; Mancini et al., 2010) has been used as a parallel of velocity when multiple points are measured. Other option is the calculation of statistical features of acceleration, velocity and jerk (Fabiani et al., 2012). Based on expressiveness movement analysis theories, some authors have chosen specific features which provide abstract gestural information. For example, *contraction* has been used to estimate body changes in space (Maes et al., 2010; Gillian & Paradiso, 2012). Mancini et al. (2010) proposes the usage of *impulsivity*, defined as the ratio between energy and duration of a movement, and *directness*, a measure of the degree of flexibility of a trajectory.

Another approach is the recognition of what gesture the musician is doing. This type of approximation typically involves the application of recurrent neural networks (RNNs) or hidden Markov models (HMM) in order to classify which gesture the user is doing on a set of pre-learned gestures. In this case, a set of gestures needs to be defined and the system is trained to work just with these movements. Typically, users can add their own gestures to the dataset of learned gestures. Software tools such as the Wekinator (Fiebrink & Caramiaux, 2016) or x2Gesture (Street et al., 2016), have been developed with the aim of facilitating the musicians' work in gesture recognition.

The usage of a theoretical framework for the understanding of gesture expression, such as Laban's theories (Laban & Ullmann, 1975), allows for a higher level of analysis. The Laban movement analysis (LMA) system provide a framework to understand movements in terms of the usage of space and expressiveness (Truong et al., 2016; Hachimura et al., 2005; Larboulette & Gibet, 2015; Loke et al., 2007). Laban defined four qualities for movement description: body, space, shape and effort. *Body* describes states and spatial relations between the body parts. The *space* quality refers to the path followed by the movement in terms of direction and use of the space. *Shape* is a description of the geometrical evolution of the movement in time. Finally, *effort* describes the behavior of energy and dynamics in the development of the movement, which is related with expressiveness and style (Samadani et al., 2013; Larboulette & Gibet, 2015).

Based on implementations of LMA, some DMIs map expressive features of gestures to synthesis variables (Loke et al., 2007; Fenza et al., 2005; Visi et al., 2017). As Labanotation was initially developed for dance, implementations are conceived for full-body systems and the creation of sounds from dancers' movements. In this context, the effort and shape qualities have usually been considered more relevant. Body and space qualities are related with the kind of movement being made, while effort and shape, instead, describe how the movement is made. For these reasons, effort and shape are typically used to describe expressive movements, as they give information about expressiveness and style in the performance of a gesture (Samadani et al., 2013; Fenza et al., 2005; Karg et al., 2013).

3.2.2. Expressive analysis of gestures in multi-touch DMIs

The incorporation of expressiveness to DMIs with multi-touch surfaces presents particular challenges for instrument designers, because full-body analysis descriptions can't be directly applied. Hence, we now describe some efforts that have been made in order to include expressiveness in the processing stages of DMIs with touchable interfaces.

One approach is the usage of descriptors based on individual fingers movements. For example, the Linnstrument (Linn & Roger Linn Design, n.d.) and the Seaboard (Roli, 2018) calculate each finger movement's descriptors separately in order to relate them with the later synthesis process. Each finger is associated with a MIDI voice and their features are modified in terms of the descriptors that each instrument calculates. In these cases, 3D position and velocities are relevant but also, some specific changes of gestures are calculated. The Linnstrument and Seaboard use similar approaches for mapping: pressure is related with loudness, pitch is set in a fixed position but modified by left/right movements and timbral properties are modified with forward and backward movements. Also, strike and release velocity are used to modify the timbral properties at the end and beginning of the sound. The Orphion (Trump & Bullock, 2014) and GeoShred app (Wizdom Music, n.d.), are other examples, where deviations from a pre-defined zone are used as triggers for expression.

A different idea is to obtain expressive features from the relations between different fingers. For example, the apps TC-11 (Bit Shape, n.d.-b) and TC-data (Bit Shape, n.d.-a) calculate descriptors for fingers movements but also include descriptors of relations between different fingers that are touching the surface (Schlei, 2012, 2015). Angles and distances can also be used to control different variables of the sound synthesis process. Also, the device (typically a tablet) motion sensors are typically incorporated to get information from its accelerometer or gyroscope.

3.3. Gestural mapping

Mapping is a central stage in the process of sound production of a DMI and it is considered a key part for the expressiveness of musical instruments. As we established before, data obtained from the interaction and processing stages has to be mapped into sound synthesis parameters. However, as working with gestural information typically implies working with a large number of inputs, the mapping strategy is not easy to design.

The usage of metaphors is considered a desirable feature in a mapping process. On one side, human-computer interaction studies of DMIs have shown the importance of a clear understanding about the instrument functioning: when musicians understand how to modify the sound, the interaction performer-instrument works (Fels et al., 2002). On another side, embodied cognition studies have highlighted the benefits of the usage of metaphors in interactive interfaces and determined that they facilitate learning and usability (Antle et al., 2009; Bakker et al., 2012). In summary, a mapping stage based on metaphors can help designers to simplify how the instrument work and achieve a better performance.

Different strategies for the mapping stage can modify the complexity of the sound production stage. For example, a mapping can take only one movement variable and translate it into one synthesis variable, called a one-to-one strategy. A one-to-many strategy relates one input to multiple outputs. Many-to-many strategies, meanwhile, connect changes in multiple parameters of movement with multiple parameters of sound resulting in several cross-coupling relations (Wanderley & Depalle, 2004). Research has shown that one-to-one and one-to-many strategies limit the interaction and expressiveness due to its simplicity: musicians often think about how the synthesis work instead of how to play the instrument to achieve a specific sound (Wanderley & Depalle, 2004). Many-to-many strategies, on the contrary, can create complex mappings that allow musicians to control the sound in multiple ways, as it usually happens in traditional instruments, and, as a consequence, produce more expressive results (Hunt et al., 2003; Wessel & Wright, 2002).

Some systems use mathematical functions to map gestural information into synthesis parameters. This includes the usage of gesture descriptors in the processing stage: mathematical expressions that describe some aspects of movements. Descriptors are taken as the inputs for a mathematical function which determines the value of a particular synthesis variable. The function can be a well-known mathematical expression or can be designed for this specific purpose. Functions typically must be fine-tuned to work with specific parameters, and they can be pre-designed, but in many cases they are open to let the musicians modify the instrument as they wish, as in (Maes et al., 2010; Perrotin & D'Alessandro, 2013). One problem of this approximation is the difficulty of generating interesting multi-parametric functions which could allow many-to-many mappings. As a result, parallel one-to-one strategies are common, negatively impacting the creation of metaphors that involve change of a large number of parameters at the same time.

There are also systems that define mapping rules of behavior. Typically, this implies the definition of multiple rules, based on expert knowledge, that are activated according to the descriptor values. Some approximations work with decision trees that select different rules of behavior following the previous decisions. This approach can be mixed with customized functions or representations of expressiveness in different levels of mapping.

Rules can imply settings about parameters and usually require many rules and a significant amount of previous analysis.

The usage of recognition systems, based on machine learning techniques, imply a fixed mapping stage modified with specific descriptors. Once a gesture is recognized, it is associated with a specific sound, making it necessary to train the system by selecting some gestures and sounds that constitute the base of the system (Caramiaux et al., 2014; Paradiso, 2014). This approach has problems adding variations to the sound synthesis, as subtle changes on movements would not necessarily be captured directly. As a result, expressiveness is constrained, because sounds can not be finely modified. To add expressive features, some DMIs have incorporated other descriptors of movement. For example, the position (Geiger, 2006) and the variance from the learned gestures (Street et al., 2016) can be used to achieve a greater level of expression.

The usage of fuzzy logic is another possible strategy which gives the option of mapping many inputs into multiple outputs based on simple linguistic rules (Kreković & Pošćić, 2015; R. F. Cádiz & Gonzalez-Inostroza, 2018). Fuzzy logic helps to treat the variables of a system in a conceptual way, instead of a numerical one, which is easier and more attuned to use in musical contexts. Also, it allows the usage of movement descriptors with multiple outputs and complex rules. Rules can define a behavior based on the understanding of the movements and their features, as we will discuss now.

3.3.1. Fuzzy-based mapping

Fuzzy logic (Bandemer & Gottwald, 1995; Cox, 1994; Klir & Yuan, 1995; Kosko, 1993; McNeill & Freiberger, 1993) is a concept derived from the mathematical branch of fuzzy sets (Zadeh, 1965) that applies multi-valued logic to these sets. In a narrow sense, fuzzy logic refers to a logical system that generalizes traditional two-valued logic for reasoning under uncertainty, allowing multiple values of truth. In a broader sense, it refers to all theories and technologies that employ fuzzy sets (Yen & Langari, 1999). In general, when fuzzy logic is applied to a problem, certain aspects of the human reasoning process can be emulated, imprecise information can be taken into account and decisions based on vague and incomplete data can be made (Kosko, 1993).

Fuzzy logic systems have been widely used in a variety of fields, most prominently engineering and control applications (Kosko, 1993) (Klir & Yuan, 1995), but they have also been applied to other areas as diverse as economics, business and finance (Von Altrock, 1997), sociology (Dimitrov & Hodge, 2002) and geology (Demicco & Klir, 2004). In the specific case of music, Landy included fuzzy logic as one of the potential areas for the future music world (Landy, 2001). Fuzzy logic has been used in the computer music field to compose, control and synthesize sound. As "fuzzy logic is a powerful way to implement non-linear mappings and intuitive control of a high number of non-intuitive synthesis parameters" (R. F. Cádiz & Gonzalez-Inostroza, 2018), it is an ideal approach for DMI mapping design.

In a fuzzy model, variables are treated as "fuzzy" variables, in the sense that they represent a conceptual understanding of a feature instead of a pure numerical one, and rules can be easily specified in the form of IF-THEN statements. This facilitates the task of designing a complex and non-linear mapping strategy. In order to facilitate the work of musicians and DMI designer with fuzzy logic systems, the authors have developed a software tool labelled the Fuzzy Logic Control Toolkit (FLCTK) (R. F. Cádiz & Gonzalez-Inostroza, 2018; R. Cádiz & Kendall, 2006), a series of functions implemented as external objects for the widely used real-time compositional environments Max/MSP (Cycling '74, n.d.) and Pd (Pd Community, n.d.). This tool allows to create and modify fuzzy systems in the fly and try its behavior in real-time.

3.4. A system for expressive sound control in multi-touch DMIs

We believe that a deeper understanding, focused on expression, of the fingers' movements would allow us to make more expressive and natural sound synthesis for multi-touch environments. In order to do that, we present a system which helps to control with a high

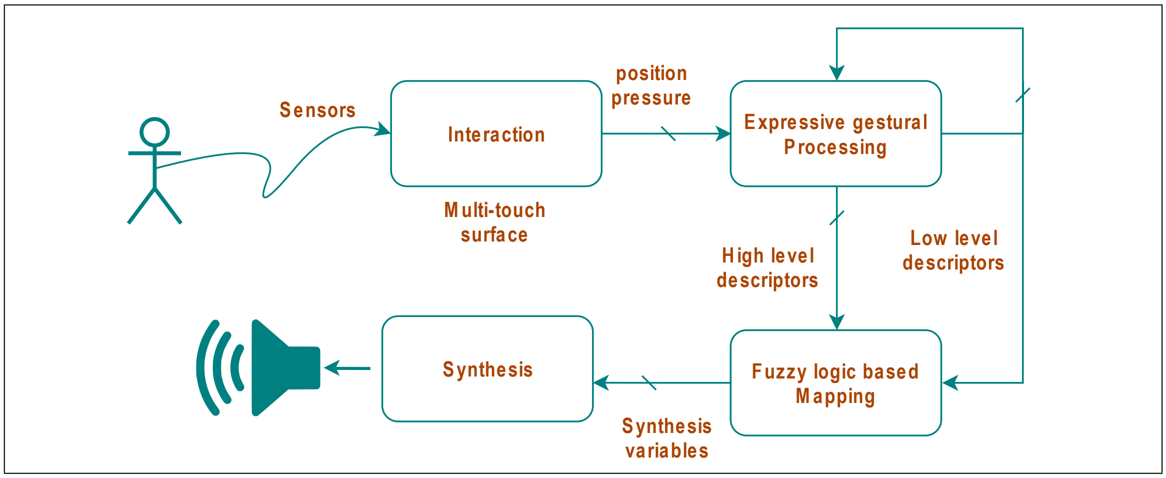


Figure 3.2. DMIs model applied to the proposal. Our system is designed for DMIs using multi-touch surfaces. The processing stage calculates expressive gestural descriptors, which are the inputs for the mapping stage that use a fuzzy logic inference engine to map descriptors into synthesis variables following the rules defined by the user.

degree of detail a sound synthesis engine based on an expressive analysis of gestures, using expressive gestural descriptors and a fuzzy logic system. The descriptors of section 3.4.1 help us to understand gestures in terms of expressiveness and geometry, letting the user define rules that links the musician expressiveness with the sound qualities. In the mapping stage, a fuzzy logic system maps the descriptors (inputs) to synthesis parameters (outputs) following rules that the musician/composer can specify as will. The rules of the fuzzy system can be easily modified in order to fine-tune a specific task.

An interaction stage for a general multi-touch surface, that provides information about the position and the pressure of each finger, was considered. In the processing stage, as figure 3.2 illustrates, this information is used to estimate finger descriptors that give information about geometry and expressiveness features of the gestures being produced. There is a set of 19 general descriptors and 25 per finger, that users can choose in order to generate rules that describe the gestures that they consider appropriate. The mapping stage consists of a fuzzy system that links the descriptors with synthesis variables following the rules defined by the user. In order to facilitate the work, inputs and outputs were normalized. The scaled outputs of the fuzzy system can be connected, as inputs, to any synthesis system. Finally, when the musician plays the DMI, the synthesis variables will be updated according to the rules.

Our system is implemented in MaxMSP and can work with any multi-touch surface. Some descriptors are calculated for each finger and others depend of the interaction between all of them. The FLCTK (R. F. Cádiz & Gonzalez-Inostroza, 2018) was used to implement the fuzzy logic engine. The system receives the inputs and gives the outputs following the specified rules. This strategy allows for rule-based mappings that consider not just geometry estimations, but also expressiveness. The rules can be easily made to represent metaphors, helping musicians and audiences to understand in more clarity the inner working of the DMI while being performed on stage.

3.4.1. Expressive gestural description on multi-touch surfaces

In the following section we describe a series of descriptors that capture expressive features of the fingers while they move along a touchable surface. Despite previous efforts to study hand movements from an expressive perspective (Samadani et al., 2011; Laiyang & Junjun, 2014), the study of gestures in multi-touch surfaces is not abundant and it has mostly focused on positions and dynamics. Based on Larboulette & Gibet (2015), we selected some low and high level descriptors used in classic analysis of gestures and we incorporated others that are more connected to expressiveness. Most of them have been typically applied to all-body movements, so we adapted them for the case of multi-touch environments. Some descriptors were calculated for each finger k and others were calculated for all the fingers that are detected at a given time. The group which involved every finger k is denoted with the letter K.

3.4.1.1. Low level descriptors

These descriptors give information that can be calculated very fast and describe basic features of movements. We classify the descriptors as individual or grouped, in order to analyze the movement of a single finger in contrast with the contextual movement of all the fingers. The following formulas use a non-causal notation, but they can be properly adjusted when implemented in real-time.

- Individual descriptors:
  - **position:** vectorial description of the 3D position of finger k

$$\vec{p_k}(t_i) = \begin{pmatrix} x_k(t_i) \\ y_k(t_i) \\ z_k(t_i) \end{pmatrix}$$

  - **displacement:** vectorial description of the 3D displacement of the fingertip k from its position at the beginning of the movement.

$$\vec{d_k}(t_i) = \vec{p_k}(t_i) - \vec{p_k}(0)$$

  - **velocity:** change in time of the position

$$\vec{v_k}(t_i) = \frac{\vec{p_k}(t_{i+1}) - \vec{p_k}(t_{i-1})}{2t}$$

  - **acceleration:** change in time of the velocity

$$\vec{a_k}(t_i) = \frac{\vec{p_k}(t_{i+1}) - 2\vec{p_k}(t_i) + \vec{p_k}(t_{i-1})}{t^2}$$

  - **jerk:** the derivative of acceleration, related with smoothness of a movement

$$\vec{j_k}(t_i) = \frac{\vec{p_k}(t_{i+2}) - 2\vec{p_k}(t_{i+1}) + 2\vec{p_k}(t_{i-1}) - \vec{p_k}(t_{i-2})}{2t^3}$$

  - **curvature:** the inverse of the radius of a trajectory at time t_i

$$C_k(t_i) = \frac{\|\vec{a_k}(t_i) \times \vec{v_k}(t_i)\|}{\|\vec{v_k}(t_i)\|^3}$$

  - **time since origin:** milliseconds since a finger began to touch the surface
- Group descriptors:
  - **number of fingers:** number of fingers that are touching the surface

- **total area:** the sum of the areas, related to pressure, of all active fingers

$$A_T = \sum_{k \in K} A_k(t_i)$$

- **center of mass:** weighted average position considering the areas of the fingers

$$CoM(t_i) = \frac{\sum_{k \in K} A_k(t_i) * \vec{p_k}(t_i)}{A_T}$$

- **bounding box:** volume that spatially encapsulates the movement.

  The descriptors that were calculated for individual fingers are also calculated for groups considering the weighted average of active fingers. As the formulas are very similar, they are not explicitly shown here.

3.4.1.2. High level descriptors

We propose the usage of descriptors based on LMA, but adapted to fingers in order to analyze the movements of the musicians and subsequently map them into synthesis variables. As was mentioned in section 3.2.1, the effort and shape features of movement are connected to expressiveness and we focus on them both. Some mathematical implementations of effort and shape for hands and fingers have been proposed (Samadani et al., 2011, 2013; Laiyang & Junjun, 2014). Typically they consider different parts of the hand as single points in space. Consequently, descriptors are calculated for a single point or for a group of them. In multi-touch systems, we just have information about the positions of fingers that are touching the surface. In this paper we present some adaptations of effort and shape dimensions of LMA to our context. In the following formulas, we assume that a generic multi-touch surface accurately provides the position and an estimation of pressure of each finger in 3D and a unique ID to identify each one.

- **Effort descriptors** describe the use of energy and expressiveness in the development of the movement. They are divided into four factors: space, time, flow and weight.
  - **Weight effort** focuses on the usage of energy. Two opposite motions can be described in terms of their weight: strong or light. A strong movement has a higher level of weight effort, as it is powerful and involves a greater force amount. Light movements, on the contrary, have lower levels of weight as they use less energy and are more delicate.

$$Weight_k(T) = max\|\vec{v_k}(t_i)\|^2, i \in [1, T]$$

$$Weight(T) = max \sum_{k \in K} w_k \|\vec{v_k}(t_i)\|^2, i \in [1, T]$$

  - **Time** describes a sense of urgency of the movement: sudden or sustained. Sudden movements are quickly and urgent, and have a bigger value of time effort descriptor. Sustained movements are continuous, with small changes in time, resulting in a smaller time effort.

$$Time_k(T) = \frac{1}{T} \sum_{i=1}^{T} \|\vec{a_k}(t_i)\|$$

$$Time(T) = \frac{1}{T} \sum_{k\ inK} \sum_{i=1}^{T} w_k \|\vec{a_k}(t_i)\|$$

  - **Space** express how the space is used and how the movement is related with its surroundings. It differentiates two opposite poles: direct (straight, focused) and indirect (flexible, multi-focused).

$$Space_k(T) = \frac{\sum_{i=2}^{T} \|\vec{x_k}(t_i) - \vec{x_k}(t_i - 1)\|}{\|x_k(T) - x_k(t1)\|}$$

$$Space(T) = \sum_{k\ inK} w_k \frac{\sum_{i=2}^{T} \|\vec{x_k}(t_i) - \vec{x_k}(t_i - 1)\|}{\|x_k(T) - x_k(t1)\|}$$

Direct movements have a smaller value of Space effort than indirect movements.

– **Flow** describes the continuity of the movement. It distinguishes between free and bounded movements.

$$Flow_k(T) = \frac{1}{T} \sum_{i=1}^{T} \|\vec{j_k}(t_i)\|$$

$$Flow(T) = \frac{1}{T} \sum_{k\ inK} \sum_{i=1}^{T} w_k \|\vec{j_k}(t_i)\|$$

A free movement has a higher value of flow effort compared to a bounded one.

- **Shape descriptors** show how the body, or fingers in our case, are changing in time. We apply it to relations between different touch points.

  – **Directional** indicates how the path that the movement follows is. It distinguishes between arc-like and a spoke-like trajectories.

$$Dir(T) = \frac{1}{T} \sum_{k\ inK} w_k \sum_{i=1}^{T} C_k(t_i)$$

A high directional value implies arc-like movements.

– **Shape flow** describes the evolution of the relationships between fingers. Maximum and minimum values were previously calculated in the bounding box.

$$A_x(t_i) = \|Max_x(t_i) - Min_x(t_i)\|$$

We use the volume covered by the fingers:

$$Volume(t_i) = A_x(t_i) * A_y(t_i) * A_z(t_i)$$

To obtain the shape flow, we must calculate the change of the volume:

$$Shape(t_i) = \frac{Volume(t_i) - Volume(t_{i-1})}{t_i - t_{i-1}}$$

– **Shaping** specifies how the shape is changing in terms of space usage in each dimension. For dimension x we have:

$$Shaping_x(t_i) = \frac{A_x(t_i) - A_x(t_{i-1})}{t_i - t_{i-1}}$$

The same equation could be applied for each dimension. It let us distinguish between rising/sinking, retreating/advancing and enclosing/spreading changes in each corresponding dimension. A positive value indicates a growth in the respective dimension and a negative value a shrinking.

3.4.1.3. High level descriptors

We propose the usage of descriptors based on LMA, but adapted to fingers in order to analyze the movements of the musicians and subsequently map them into synthesis variables. As was mentioned in section 3.2, the effort and shape features of movement are connected to expressiveness and we focus on them both. Some mathematical implementations of effort and shape for hands have been proposed (Samadani et al., 2011; Laiyang & Junjun, 2014). Typically they consider the whole hand as a single point in space. Consequently, descriptors are calculated for a single point or for a group of them, but not for individual fingers. In this paper we present some adaptations of effort and shape dimensions of LMA to our context. In the following formulas, we assume that a generic multi-touch surface accurately provides the position and an estimation of pressure of each finger in 3D and a unique ID to identify each one.

- **Effort descriptors** describe the use of energy and expressiveness in the development of the movement. They are divided into four factors: space, time, flow and weight.
  - **Weight effort** focuses on the usage of energy. Two opposite motions can be described in terms of their weight: strong or light. A strong movement has a higher level of weight effort, as it is powerful and involves a greater force amount. Light movements, on the contrary, have lower levels of weight as they use less energy and are more delicate.

$$Weight_k(T) = max\|\vec{v_k}(t_i)\|^2, i \in [1, T] \tag{3.1}$$

$$Weight(T) = max\sum_{k \in K} w_k\|\vec{v_k}(t_i)\|^2, i \in [1, T] \tag{3.2}$$

  - **Time** describes a sense of urgency of the movement: sudden or sustained. Sudden movements are quickly and urgent, and have a bigger value of time effort descriptor. Sustained movements are continuous, with small changes in time, resulting in a smaller time effort.

$$Time_k(T) = \frac{1}{T}\sum_{i=1}^{T}\|\vec{a_k}(t_i)\| \tag{3.3}$$

$$Time(T) = \frac{1}{T}\sum_{k\ inK}\sum_{i=1}^{T} w_k\|\vec{a_k}(t_i)\| \tag{3.4}$$

  - **Space** express how the space is used and how the movement is related with its surroundings. It differentiates two opposite poles: direct (straight, focused) and indirect (flexible, multi-focused).

$$Space_k(T) = \frac{\sum_{i=2}^{T}\|\vec{x_k}(t_i) - \vec{x_k}(t_i - 1)\|}{\|x_k(T) - x_k(t1)\|} \tag{3.5}$$

$$Space(T) = \sum_{k\ inK} w_k \frac{\sum_{i=2}^{T} \|\vec{x_k}(t_i) - \vec{x_k}(t_i - 1)\|}{\|x_k(T) - x_k(t1)\|} \tag{3.6}$$

Direct movements have a smaller value of Space effort than indirect movements.

– **Flow** describes the continuity of the movement. It distinguishes between free and bounded movements.

$$Flow_k(T) = \frac{1}{T} \sum_{i=1}^{T} \|\vec{j_k}(t_i)\| \tag{3.7}$$

$$Flow(T) = \frac{1}{T} \sum_{k\ inK} \sum_{i=1}^{T} w_k \|\vec{j_k}(t_i)\| \tag{3.8}$$

A free movement has a higher value of flow effort compared to a bounded one.

- **Shape descriptors** show how the body, or fingers in our case, are changing in time. We apply it to relations between different touch points.

  – **Directional** indicates how the path that the movement follows is. It distinguishes between arc-like and a spoke-like trajectories.

$$Dir(T) = \frac{1}{T} \sum_{k\ inK} w_k \sum_{i=1}^{T} C_k(t_i) \tag{3.9}$$

A high curvature value implies arc-like movements.

  – **Shape flow** describes the evolution of the relationships between fingers. Maximum and minimum values were previously calculated in the bounding box.

$$A_x(t_i) = \|Max_x(t_i) - Min_x(t_i)\| \tag{3.10}$$

We use the volume covered by the fingers:

$$Volume(t_i) = A_x(t_i) * A_y(t_i) * A_z(t_i) \tag{3.11}$$

To obtain the shape flow, we must calculate the change of the volume:

$$Shape(t_i) = \frac{Volume(t_i) - Volume(t_{i-1})}{t_i - t_{i-1}} \tag{3.12}$$

– **Shaping** specifies how the shape is changing in terms of space usage in each dimension. For dimension x we have:

$$Shaping_x(t_i) = \frac{A_x(t_i) - A_x(t_{i-1})}{t_i - t_{i-1}} \tag{3.13}$$

The same equation could be applied for each dimension. It let us distinguish between rising/sinking, retreating/advancing and enclosing/spreading changes in each corresponding dimension. A positive value indicates a growth in the respective dimension and a negative value a diminution.

3.5. Arcontinuo: a case of study

Figure 3.3. The Arcontinuo (R. F. Cádiz & Sylleros, 2017) is a DMI with a curved multi-touch surface. It provides a continuous tracking of all ten fingers in three-dimensional space.

In order to test our proposed fuzzy gestural mapping approach, the Arcontinuo was chosen. Arcontinuo is a DMI with a user-centered design, consisting on a curved multi-touch surface, as it can be seen in figure 3.3, capable of detecting and tracking positions and pressures of all ten fingers (R. F. Cádiz & Sylleros, 2017; Sylleros et al., 2014). The Arcontinuo outputs the finger information via MIDI Polyphonic Expression (MPE), making it compatible to almost any sound synthesis engine on the market.

3.5.1. Test of expressive gestural descriptors

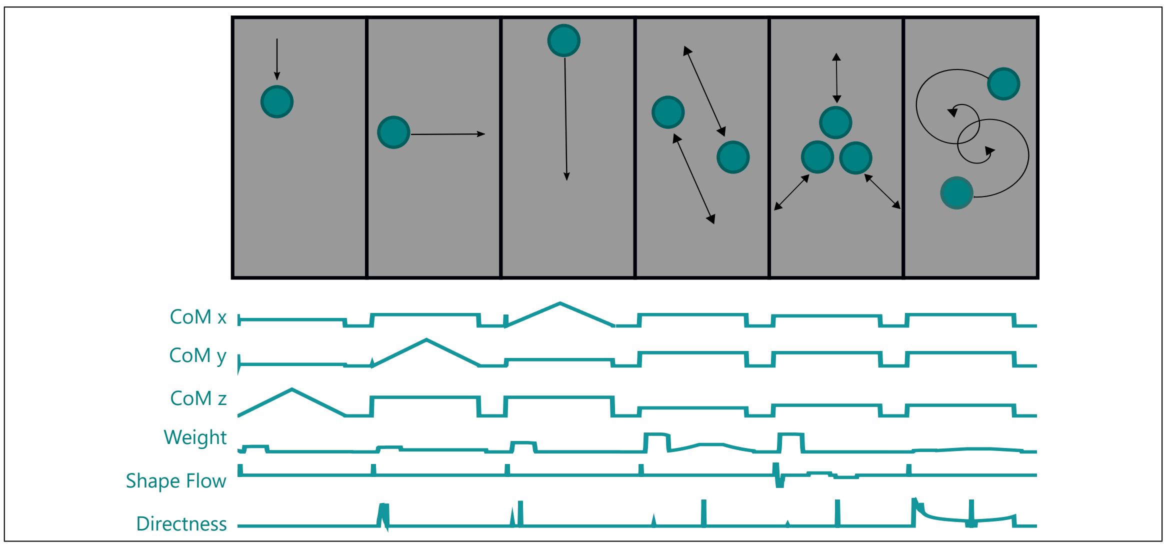


Figure 3.4. Descriptors' behavior for simulated Arcontinuo. The center of mass in x, y and z, weight, shape flow and directness change following the changes of movements (top). In the three first cases, the simulation just changed the center of mass in one of the axes. In the latter cases even when center of mass does not change, descriptors related with expressiveness do change.

We implemented the descriptors of section 3.4.1 in MaxMsp (Cycling '74, n.d.) and tested them with the Arcontinuo and a simulated version of it. The inputs are position (x,y), pressure (z) and area for each finger. We use overlapped windows of 15 samples for descriptors that consider more than one sample, but the user can set how many samples the system takes into account. Another variable to take into consideration is the weighting of each finger for group descriptors. Other implementations of gestural descriptors have

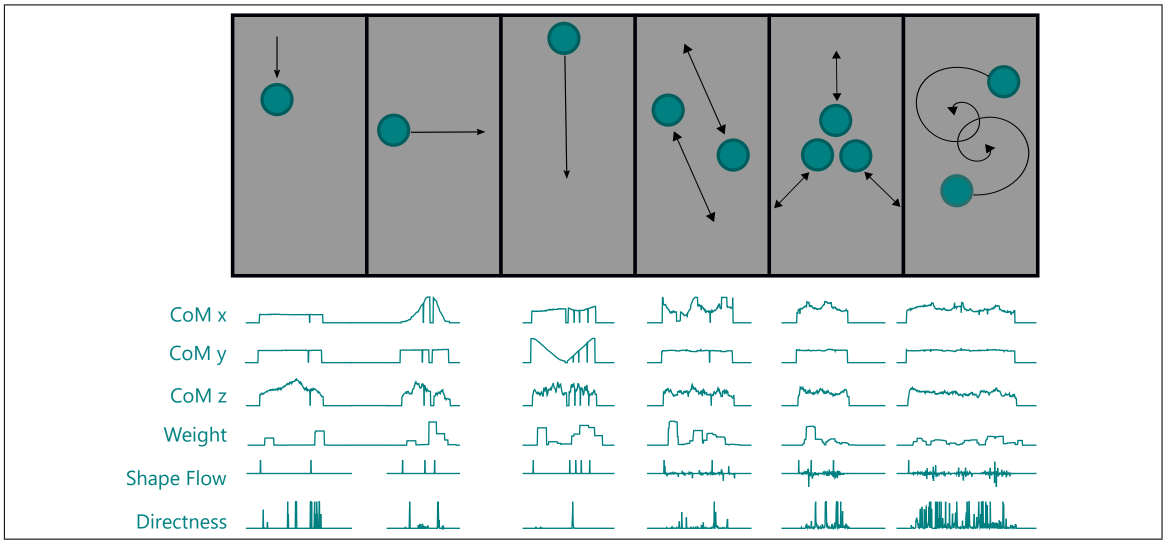


Figure 3.5. Descriptors' behavior for real Arcontinuo. The center of mass in x, y and z, weight, shape flow and directness change following the changes of musician's gestures (top). The descriptors behaviors are similar to the simulated version.

proposed a fixed value depending on the volume of each part of the body or one for every analyzed part. For the case of multi-touch surfaces, as the number of fingers touching the surface is not fixed and it's not possible to detect the actual finger that is acting, it is important to consider other options. In our system, the weight of descriptors for each finger correspond to its area divided by the sum of all the areas.

We compared different movements and the changes in descriptors they made. Descriptors actually give information that let us distinguish between them. As an example, in figures 3.4 and 3.5 we compare descriptors values in a simulated version of Arcontinuo and the real one for a set of gestures: press and release, horizontal and vertical trajectories, two fingers moving diagonally, three fingers enlarging and reducing the triangle they form, two fingers drawing circles with variable radius. Each gesture principally affect one descriptor for both the simulated and real versions of the Arcontinuo. We observed that even though the last three gestures conserve their center of mass, there are descriptors that capture their changes rapidly. For example, in the fifth gesture, when the shape is growing and shrinking but the center of mass is the same, nevertheless, we can rely on shape flow to capture this change.

3.5.2. Mapping based on fuzzy logic

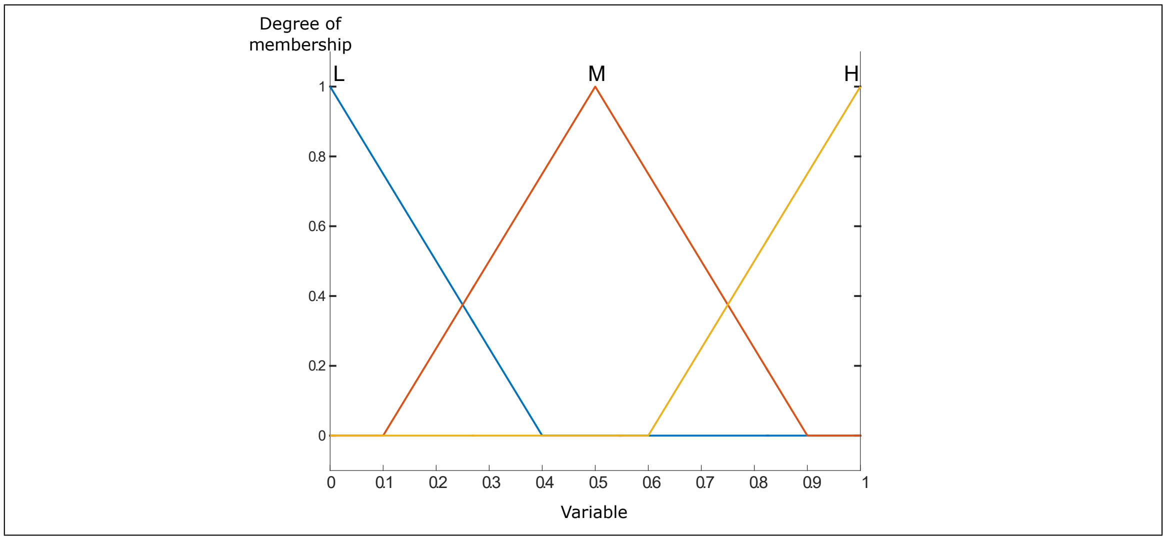


Figure 3.6. Membership functions used for fuzzy inputs and outputs have triangular shapes. Values are classified in three categories: L (low), M (medium) and H (high). As it is a fuzzy representation, variables can belong to more than one membership function at the same time.

Using a fuzzy system implies the definition of inputs, outputs and rules. We defined six outputs and selected six inputs that belong to some of the gestures descriptors explained in section 3.4.1: Center of mass in x, y, z, weight effort, shape flow and directness. We considered normalized versions of the inputs and outputs in order to simplify the fuzzification. Three fuzzy membership functions, with a triangular shape were chosen, as figure 3.6 displays, in order to define the degree of membership for each input and output: low (L), medium (M) and high (H). These three levels correspond to a conceptual understanding of descriptors and can be related to a qualitative description of gestures. For example, a slow gesture has a low velocity. In the case of descriptors adapted from Labanotation, as table 3.1 shows, low and high functions correspond to a specific feature of gestures. For example, a low shape flow is equivalent to a shrinking gesture.

Table 3.1. Equivalences of fuzzy memberships for Labanotation-derived descriptors. Laban defined opposite gestures for each dimension, and they can be associated with the values of their corresponding descriptors. For example, a movement can be bounded or free; if it is bounded, the flow effort is low and if it is free the flow effort is high.

Descriptor	Low (L)	High (H)
Weight Effort	light	strong
Time Effort	sustained	sudden
Space Effort	direct	indirect
Flow Effort	bounded	free
Directional	spoke-like	arc-like
Shape flow	shrinking	growing
Shaping x	widening	narrowing
Shaping y	lengthening	shortening
Shaping z	bulging	hollowing

We defined the rules following three strategies that show different possibilities of expressive mappings. The first case, discussed in section 3.5.2.1, considered many one-to-one rules, built based on one descriptor related to only one output variable. The example described in section 3.5.2.2 encompasses rules based on some complex gestures, where changes in more than one descriptor affect two or more outputs. Finally, the third example, presented in section 3.5.2.3, shows how to create rules based on musical metaphors with a higher degree of complexity. In all of these cases, we hope to clarify the potential usage for our system and how it can help musicians to easily define rich and non-linear mappings based on expressive gestural descriptors.

3.5.2.1. Example 1: rules based on descriptors

In order to make a preliminary test of our system, we defined rules that link only one descriptor to only one output. We selected six inputs: Center of mass in X, Y and Z (CoMx,CoMy,CoMz), Weight effort (Weight), shape flow (Shape) and directness (Direct). The system has six outputs variables (O1, O2, O3, O4, O5, O6) that were mapped into the amplitude of six sinusoids of different frequencies. These rules were established following

Table 3.2. Fuzzy rules used for example 1. These rules map only one gestural descriptor to one synthesis variable. Different ways to define rules were used in order to check how the outputs behavior changed.

1. If (CoMz is L) then (O1 is L) (1)
2. If (CoMz is M) then (O1 is M) (1)
3. If (CoMz is H) then (O1 is H) (1)
4. If (CoMx is L) then (O2 is H) (1)
5. If (CoMx is M) then (O2 is M) (1)
6. If (CoMx is H) then (O2 is L) (1)
7. If (CoMy is L) then (O3 is L) (1)
8. If (CoMy is M) then (O3 is M) (1)
9. If (CoMy is H) then (O3 is L) (1)
10. If (Weight is L) then (O4 is L) (1)
11. If (Weight is H) then (O4 is H) (1)
12. If (Shape is not L) then (O5 is H) (1)
13. If (Shape is not H) then (O5 is L) (1)
14. If (Direct is not L) then (O6 is L) (1)
15. If (Direct is not H) then (O6 is H) (1)

the changes of six basic gestures, showed in figures 3.4 and 3.5, which only modify one descriptor at a time.

As shown in table 3.2, rules are linguistic in nature and can be designed following different strategies. For example, rules 1-3 assign the same feature from input to output; if CoMz is high, then O1 is high. In addition, we also built rules that have the inverse result (rules 4-6) or the same result for extreme values (rules 10-11). Finally, rules 12-15 use a negation operator in the inputs, so we could check the effects of such rules in the results.

In order to test the system, we simulated both the selected basic gestures and also a set of complex gestures that involve changes in multiple descriptors, to check the outputs behavior. Figure 3.7 portraits changes in inputs and outputs for the rules of this example. As it can be seen, in the first six cases, that represent basic gestures with just one varying descriptor, the modification occurs just in one output. Meanwhile, in the last six gestures, we observe changes in the same number of outputs as inputs vary.

In figure 3.7 it can be observed that outputs are not just a linear representation of the inputs, but that they replicate the instructions provided by the rules. Also, it is easy to notice that the fuzzy tool can follow different rules independently. Figure 3.8 shows

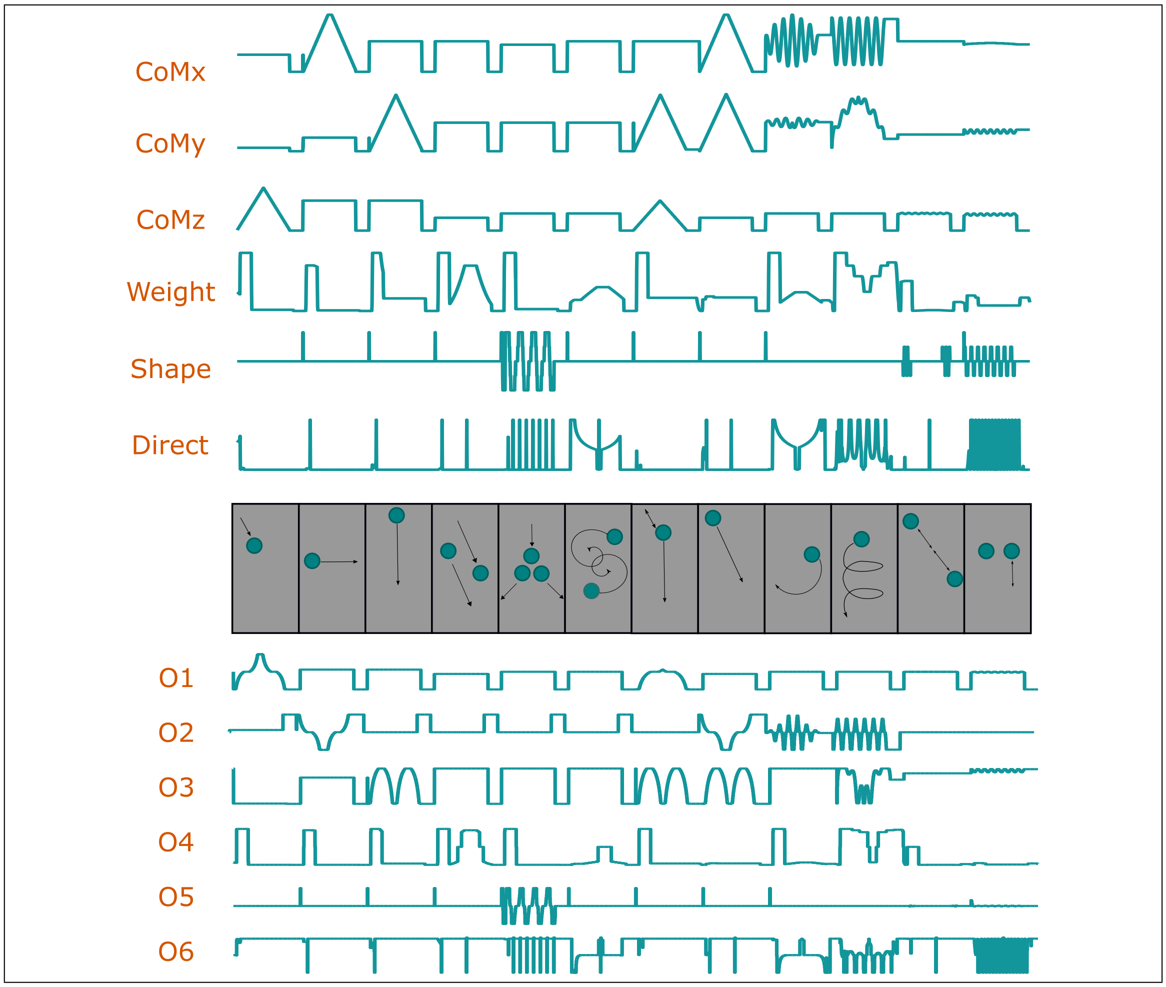
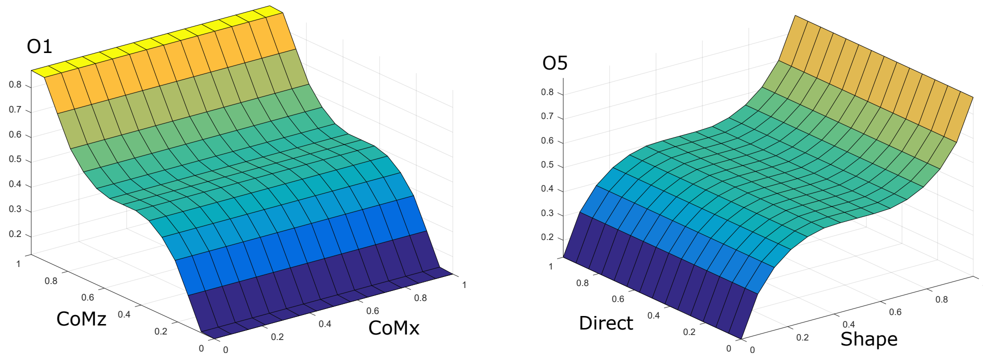


Figure 3.7. Example 1 results. Inputs (top) change according to different gestures (middle) and outputs (bottom) respond based on the rules specified in table 3.2. In this example rules modify only one output depending on only one input but they are applied all together at the same time.

clearly that the mapping depends on the rules: O1, which is connected just with regular rules, does not have a soft shape; O5, on the contrary, is related to rules with a negation operator that modify the shape and the range of the mapping surface. As rules are one-to-one, these surfaces are not very complex. A video of the results of this experiment [2] illustrates how gestures that modify just one descriptor can shape the overall sound.

[2] Available at https://drive.google.com/drive/folders/1PXj6p0jVP_pa4r3x1-UJWbkwKa8O6MYR?usp=sharing



(a) Output 1 (b) Output 5

Figure 3.8. Mapping surfaces of example 1 for the first and fifth outputs. As these rules relate just one input to one output, surfaces only change when one input does.

3.5.2.2. Example 2: rules based on a subset of gestures

Table 3.3. Fuzzy rules for example 2. Rules were constructed following descriptor values representing specific gestures. Each gesture was coupled to one behavior of a set of outputs.

1. If (CoMx is L) and (CoMy is L) then (O1 is L)(O2 is H) (1)
2. If (CoMx is M) and (CoMy is M) then (O1 is M)(O2 is M) (1)
3. If (CoMx is H) and (CoMy is H) then (O1 is H)(O2 is L)(O3 is H) (1)
4. If (CoMy is L) and (CoMz is L) then (O3 is L)(O4 is H) (1)
5. If (CoMy is M) and (CoMz is M) then (O3 is M)(O4 is M) (1)
6. If (CoMy is H) and (CoMz is H) then (O3 is L)(O4 is L) (1)
7. If (Direct is not L) and (Weight is L) then (O5 is L)(O6 is L) (1)
8. If (Direct is not L) and (Weight is not L) then (O5 is L)(O6 is H) (1)
9. If (Shape is not M) and (CoMy is H) then (O2 is L)(O4 is L)(O6 is L) (1)
10. If (Shape is not M) and (CoMy is not H) then (O2 is H)(O4 is H)(O6 is H) (1)
11. If (Direct is not L) and (CoMx is L) then (O2 is L)(O4 is L)(O6 is L) (1)
12. If (Direct is not L) and (CoMx is M) then (O2 is M)(O4 is M)(O6 is L) (1)
13. If (Direct is not L) and (CoMx is H) then (O2 is H)(O4 is H)(O6 is L) (1)
14. If (Weight is H) and (CoMz is H) then (O1 is H)(O2 is H)(O3 is H)(O4 is H)(O5 is H)(O6 is H) (1)
15. If (Weight is H) and (CoMz is not H) then (O1 is M)(O2 is H)(O3 is M)(O4 is H)(O5 is M)(O6 is H) (1)

In this second example, we have chosen a subset of gestures that imply a particular behavior of descriptors in order to build rules that respond to these gestures with a specific behavior of outputs. We analyzed the values that descriptors take when the selected gestures are performed and defined the rules detailed in table 3.3. These rules respond with

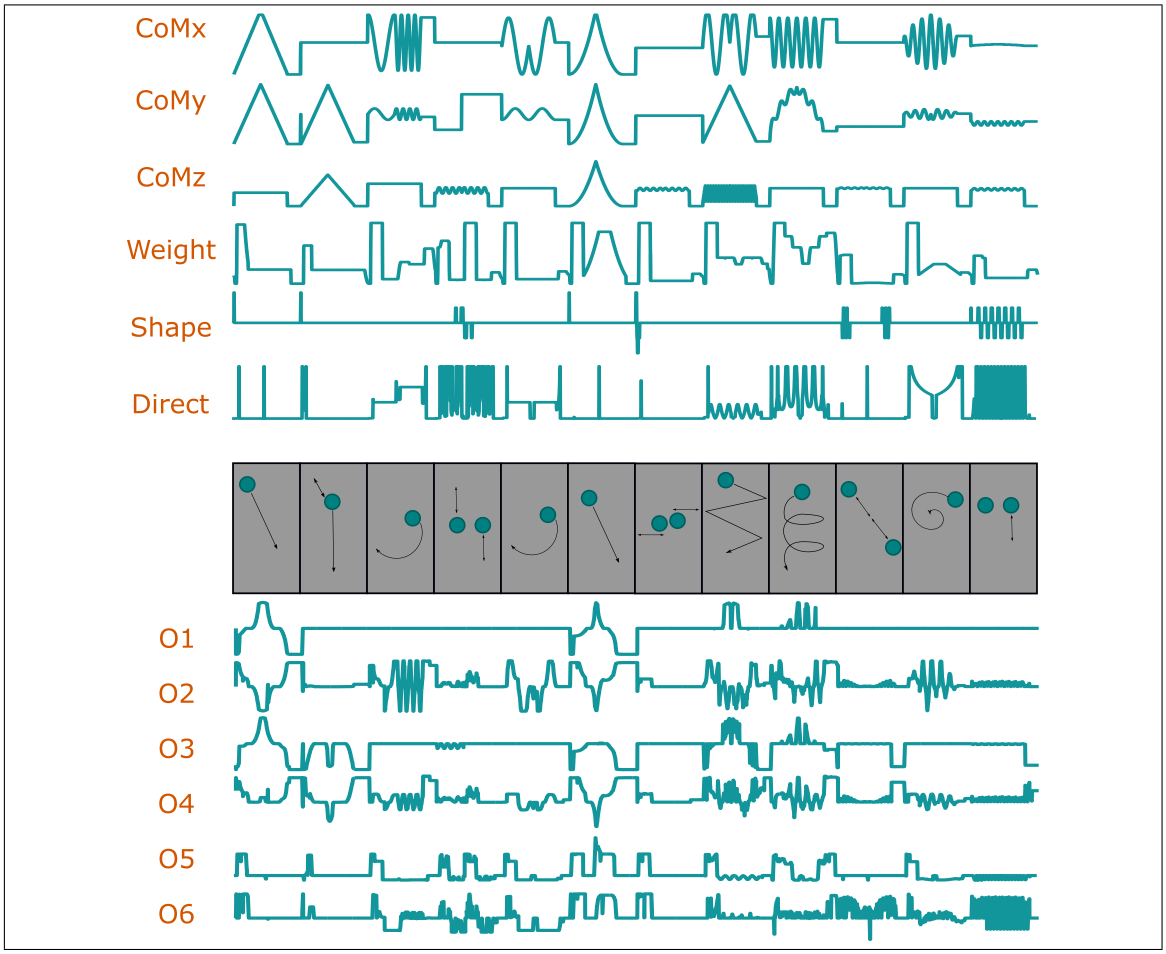


Figure 3.9. Example 2 results. Inputs (top) change following the gestures (middle) and outputs (bottom) respond according to the rules of table 3.3. In this case, rules were created using one input to represent specific gestures and they modify more than one output, but they are applied all together at the same time.

some desired values for pairs of outputs. The gestures that we considered were a diagonal line, a vertical line with changing pressure, a fast and slow circle, two fingers opening and closing vertically, a circular movement with regular velocity, and a diagonal line with variable pressure. We analyzed how descriptors change for each gesture and set the rules following these behaviors. For example, for the case of circles we realized that directness is medium or high, but never low, so in rules 7-9 and 11-13 we wanted to set a behavior for non-low values of this descriptor.

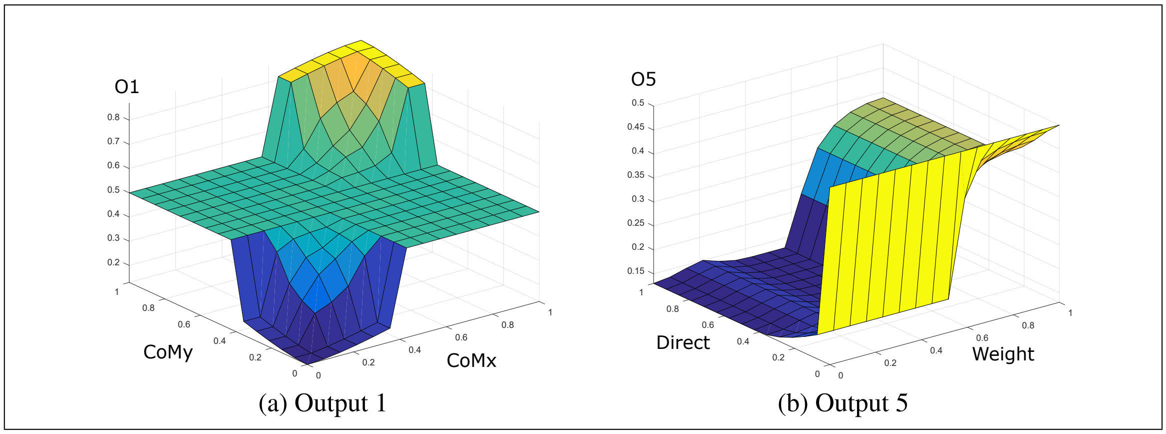


Figure 3.10. Mapping surfaces of example 2 for the first and fifth outputs.

We tested the system with the gestures captured by the rules and other simpler and complex gestures; the results are shown in figure 3.9. In this example, basic gestures successfully follow the outputs defined with the fuzzy rules, moreover, they interact creating complex mappings. The system can mix rules and react to changes that activate more than one rule, as it can be observed for diagonals and circles that activate rules 1-6 and 7, 8, 11-13 respectively. In these cases we see outputs that mix the results of all the rules that are active at a given time. Also, the system reacts satisfactorily to other gestures that were not considered in the design process, relying on the power of fuzzy systems to respond to unexpected inputs.

As shown in figure 3.10, rules act forming a coupled relation between inputs that were related in the same rules. For example, the first output, which is modified by CoMx and CoMy in rules 1-6, depends of the values of both inputs at the same time. A potential problem of this approximation is the tendency of outputs to remain in the middle of the range (0.5), as it can be checked for some outputs in figure 3.9. In the provided sound example, it is also noticeable that even when some sounds stand out from the rest, all the sounds are present all the time.

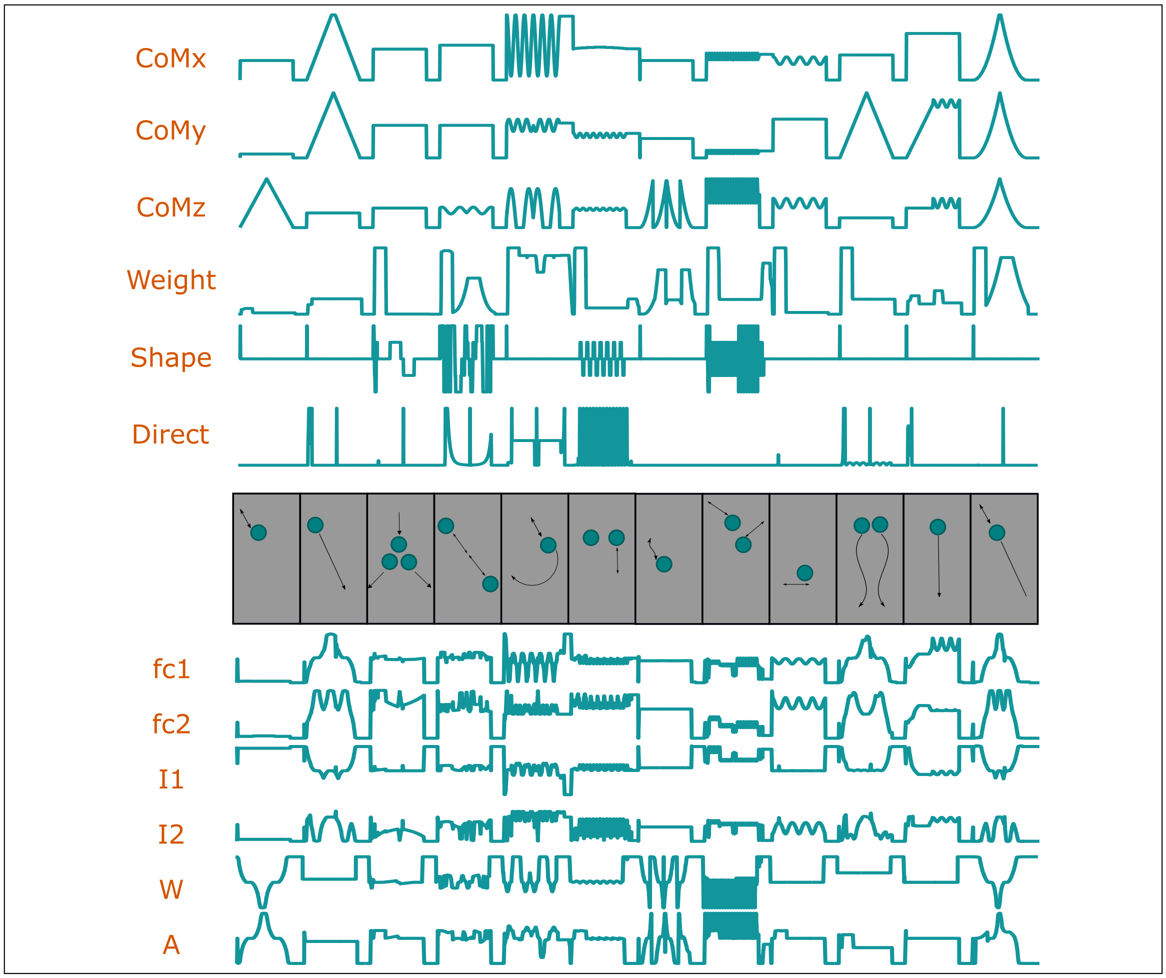


Figure 3.11. Example 3 results. Inputs (top) change following the gestures (middle) and outputs (bottom) respond according to the rules specified in table 3.4. In this case, rules were constructed following different musical metaphors. As a result, cross-coupled and complex mappings were generated.

3.5.2.3. Example 3: rules based on musical metaphors

In this example, we considered a specific musical context and generated a set of fuzzy rules that could work on a real musical performance. We connected the fuzzy system to a more complex sound synthesis engine and chose rules that could relate gestural descriptors and sets of them with specific behaviors of the sound output. We wanted to use our system to create musical metaphors in the mapping stage that could let musicians to expressively perform a multi-touch DMI.

For this example, we selected a double carrier frequency modulation (DCFM) synthesis to experiment with different possibilities in terms of sound. This synthesis follows this equation:

$$s = A_{w_1} \sin(f_{c_1} + I_1 \sin(f_{m_1})) + A_{w_2} \sin(f_{c_2} + I_2 \sin(f_{m_2}))$$

The audible result is the sum of two independent FM generators, which allows us to create formant regions in the spectrum. For this example, we considered $F_{c_1} = f_{m_1}$ and $F_{c_2} = f_{m_2}$, in order to generate harmonious sounds. To set A_{w_1} and A_{w_2}, we focused on the relations between them, so the variable W was set to define the amplitude A_{w_1} in the range [0-1] and $A_{w_2} = 1 - W$ (Roads & Strawn, 1996). The general amplitude of the sound is setted by A. The synthesis variables are then :f_{c_1}, f_{c_2}, W, I_1, I_2, A. The synthesis is finally defined as:

$$s = A(W \sin(f_{c_1} + I_1 \sin(f_{c_1})) + (1 - W) \sin(f_{c_2} + I_2 \sin(f_{c_1}))$$

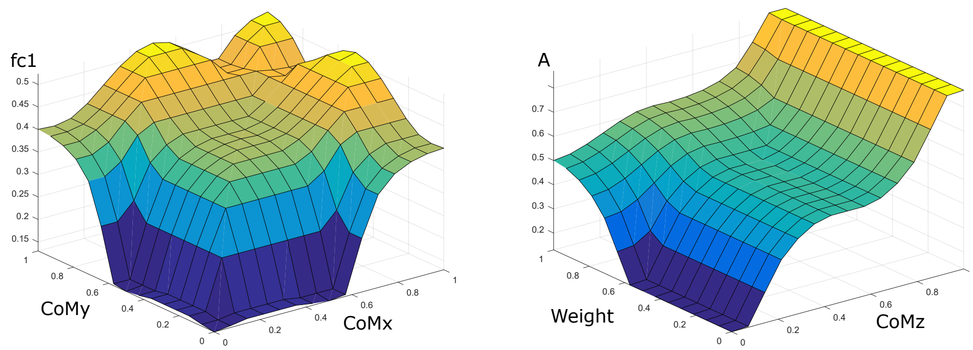
The rules were defined following some classical mapping strategies. The literature has shown that many musical instruments have a general layout to define the pitch for specific positions (or sets of them) but they can be modified with particular gestures (Hunt et al., 2003; Kvifte, 2008). Following that idea, we decided to use two kinds of rules: general rules and modifier rules. As was noted in examples 1 and 2, the usage of one-to-one rules can be used to define rules that work for one or specific groups of descriptors for a particular one gesture. In consequence, some general rules were defined with a high weight and modifier rules have lower ones. This rule design strategy can give us synthesis variables that are mostly dependant on general rules but can be subtly changed by modifier rules, giving expressiveness as the user can control these changes.

Table 3.4 shows the general and modifier rules defined for this example. The general rules (1-13) define the basic form of outputs, following a distribution of pitch and typical statements about mapping in DMIs. For example, volume is modified by the center of mass in Z, which is related with the force that the user applies, and pitch is coupled with higher frequencies at the lower and left zones of the instrument. In addition, rules 14-20 have a lower weight of 0.7 and also modify the synthesis variables, in order to give expressiveness to the system. For example, rule 14 enlarges the values of the frequency and modulation indexes when shape is not medium (so there is a big change of shape), that implies a higher pitch with a wider spectrum.

Table 3.4. Fuzzy rules for example 3. We constructed basic rules (1-12) that provide the basis for the parameters behavior but that can be changed the modifier rules (13-20), that have a lower weight. Basic rules take into account geometric descriptors and modifier rules consider expressive descriptors.

1. If (CoMz is L) then (W is H)(A is L) (1)
2. If (CoMz is M) then (W is M)(A is M) (1)
3. If (CoMz is H) then (W is L)(A is H) (1)
4. If (CoMx is L) and (CoMy is L) then (fc1 is L)(fc2 is L)(I1 is H)(I2 is L) (1)
5. If (CoMx is M) and (CoMy is L) then (fc1 is L)(fc2 is L)(I1 is H)(I2 is L) (1)
6. If (CoMx is H) and (CoMy is L) then (fc1 is M)(fc2 is M)(I1 is H)(I2 is M) (1)
7. If (CoMx is L) and (CoMy is M) then (fc1 is L)(fc2 is M)(I1 is M)(I2 is M) (1)
8. If (CoMx is M) and (CoMy is M) then (fc1 is M)(fc2 is H)(I1 is M)(I2 is L) (1)
9. If (CoMx is H) and (CoMy is M) then (fc1 is H)(fc2 is M)(I1 is L)(I2 is M) (1)
10. If (CoMx is L) and (CoMy is H) then (fc1 is M)(fc2 is H)(I1 is M)(I2 is M) (1)
11. If (CoMx is M) and (CoMy is H) then (fc1 is H)(fc2 is M)(I1 is M)(I2 is M) (1)
12. If (CoMx is H) and (CoMy is H) then (fc1 is H)(fc2 is H)(I1 is M)(I2 is M) (1)
13. If (Shape is H) or (WeightEffort is H) then (A is H) (1)
14. If (Shape is not M) and (WeightEffort is H) then (fc1 is H)(fc2 is H)(I1 is H)(I2 is H) (0.7)
15. If (Shape is not M) and (WeightEffort is M) then (fc1 is M)(fc2 is M)(I1 is H)(I2 is H) (0.7)
16. If (Shape is not M) and (WeightEffort is L) then (fc1 is L)(fc2 is L)(I1 is M)(I2 is M) (0.7)
17. If (Direct is not L) and (CoMz is H) then (fc1 is L)(fc2 is H)(I1 is H)(I2 is H) (0.7)
18. If (Direct is not L) and (CoMz is not H) then (fc1 is L)(fc2 is M)(I1 is H)(I2 is H) (0.7)
19. If (Shape is H) then (fc1 is H)(fc2 is M)(W is H) (0.7)
20. If (Shape is L) then (fc1 is H)(fc2 is L)(W is L) (0.7)

This system provides a tool to define a complex mapping based on expressive description of gestural information. We can easily define rules based on linguistic information, as table 3.4 shows, and obtain complex mappings, as it can be seen in figure 3.12. As an example, in figure 3.12a, values of fc1 (first frequency carrier), change in terms of CoMx and CoMy. We can see that frequency changes in a non-linear fashion and gives the option to find the same frequency value in different positions. Also, figure 3.12b illustrates how weight effort and CoMz interact to define the value of the amplitude of the synthesis. In this case, it is clear CoMz primarily defines the value of A, but it can be also modified by weight, especially when it has a high value. This complexity helps to give expressiveness to the system, as the sound can be modified by expressive features of gestures, giving the possibility to make subtle changes in sound based on subtle changes in gestures.



(a) Frequency carrier 1 (b) Amplitude

Figure 3.12. Surfaces show the synthesis parameters behaviors when descriptors change. Figure 3.12a illustrates values of frequency carrier 1 (fc1) for all possible values of center of mass in X and Y (CoMx and CoMy). In this case, fc1 has a complex mapping with multiple ways to obtain specific values of fc1. In figure 3.12b changes of amplitude for different values of center of mass in z (CoMz) and weight effort (Weight) can be observed. The amplitude is clearly related with the values of CoMz but high values of weight also increase it.

The testing of the system, shown in figure 3.11, revealed that the rules worked as desired. Similar gestures effectively have similar sounds that can be modified with subtle changes in the development of the gesture. For example, gestures 1 and 7, that correspond to press and release, just differ in their changes in pressure, which in turn means that the pressure changes with different velocity, which affects the weight effort. In consequence, they both keep fc1, fc2, I1 and I2 but W and A change, resulting in an expressive control of the sound. The same situation happens with diagonal lines, that differ on I2, W and A. In the case of the different tested gestures, the system responded satisfactorily following the rules, as it can be seen in the video examples.

Finally, the system was tested using a real DMI: the Arcontinuo. As it can be seen in the accompanying video (example 4)[3], as the musician performs the instrument using complex gestures, the descriptors are calculated accordingly and sent to the fuzzy inference system. The system reacts in different ways for expressive features of the gestures and, as a result, the sound is modified based on the rules previously defined in this example.

3.6. Discussion and Conclusions

How to include expressiveness in digital musical instruments is still a challenge for instrument designers. Even when technology has developed new strategies to sense and analyze movements, there is not a consensual approach about how to achieve expressiveness in DMIs. The expressive analysis of gestures can help us to develop a common understanding about emotional communication in artistic fields, with the usage of multi-modal interfaces. This lack of understanding about it is more obvious in the case of multi-touch surfaces, because most of the available studies have focused on full-body approximations. However, there are efforts that have opened the perspectives to work on stage with an expressive approximation of gestures, and we hope to contribute to that area with this work.

In order to synthesize digital sounds while performing on stage, it is important to develop a system that would let us analyze gestures at both a low and high level in real-time. Low level descriptors can explain instantaneous development of gestures, which can enlighten what is being done. High level descriptors provide a more detailed analysis in a short time about how performers are doing the gestures. In order to achieve a better understanding of gestures, the usage of the descriptors that we present in this paper can give us a numerical analysis that can be useful for designing effective mappings.

[3] Available at `https://drive.google.com/drive/folders/1PXj6p0jVP_pa4r3x1-UJWbkwKa8O6MYR?usp=sharing`

We have shown that descriptors that usually have been used on systems for full-body movement analysis also are compatible with touchable surfaces with some small modifications and provide a good approximation that would let us distinguish expressive features. Descriptors based on Laban's theories can be a great tool to work in expressive multi-touch systems: they provide some support for a mathematical analysis of expression of fingers in touchable surfaces.

Mapping of gestural information, which usually consider multiple inputs and outputs, require strategies that can let musicians modify the system's behavior easily and understand and predict how it would react. Approaches like functions settings do not provide clear information and can confuse users. On another side, approaches based on gesture recognition, such as hidden Markov models or neural networks, usually respond satisfactorily to pre-trained gestures but not to unforeseen ones. In consequence, it is desirable to have an alternative that would let us produce appealing mappings using a high level understanding of the system with appropriate responses for any gesture.

We have proposed fuzzy logic as an interesting option for DMI mapping design based on gestural information. Such a system would let us define multi-parametric common-sense rules, based on simple linguistic orders. Rules can have different degrees of relevance, that may facilitate the preponderance of some rules over others. In addition, we have shown that the incorporation of musical metaphors using our system can be easily achieved.

The design of such behavior rules can be initially made using common sense or theoretical analysis, but we consider that it is of primordial importance the incorporation of the user intention from the beginning of the process. A further study with a more acute focus on the construction of rules is therefore suggested. In this case, it would be important to test the interaction between musicians and the specific DMI being analyzed.

REFERENCES

Alessandro, N., Pon, A., Wang, J., Eagle, D., Sharlin, E., & Fels, S. (2012). A Digital Mobile Choir : Joining Two Interfaces towards Composing and Performing Collaborative Mobile Music. *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 357–360.

Alessandro, N., Tilmanne, J., Moreau, A., & Puleo, A. (2015). AirPiano : A Multi-Touch Keyboard with Hovering Control. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 255–258.

Antle, A. N., Corness, G., & Droumeva, M. (2009). What the body knows: Exploring the benefits of embodied metaphors in hybrid physical digital environments. *Interacting with Computers*, *21*(1-2), 66–75. Retrieved from `http://dx.doi.org/10.1016/j.intcom.2008.10.005`

Arcontinuo. (2015). *Arcontinuo.* Retrieved 2017-03-30, from `http://arcontinuo.com/`

Arfib, D., Couturier, J.-M., & Kessous, L. (2005). Expressiveness and Digital Musical Instrument Design. *Journal of New Music Research*, *34*(1), 125–136. Retrieved from `http://www.tandfonline.com/doi/abs/10.1080/09298210500124273?src=recsys{#}.VQHeLPmsWWs`

Artiphon. (n.d.). *Artiphon – Strum a guitar, bow a violin, tap a piano, loop a beat — on a single instrument. An intuitive digital music experience for beginners and pros alike.* Retrieved 2017-03-30, from `http://artiphon.com/`

Bailey, D. G. (2011). *Design for Embedded Image Processing on FPGAs*.

Bakker, S., Antle, A. N., & Van Den Hoven, E. (2012). Embodied metaphors in tangible interaction design. *Personal and Ubiquitous Computing*, *16*(4), 433–449.

Bandemer, H., & Gottwald, S. (1995). *Fuzzy sets, fuzzy logic, fuzzy methods with applications*. Chichester ; New York: J. Wiley.

Berndt, A., Al-Kassab, N., & Dachselt, R. (2014). TouchNoise: A Particle-based Multitouch Noise Modulation Interface. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 323–326. Retrieved from `http://www.nime.org/proceedings/2014/nime2014{_}325.pdf`

Bit Shape. (n.d.-a). *TC-11.* Retrieved 2018-09-11, from `http://www.bitshapesoftware.com/instruments/tc-11/`

Bit Shape. (n.d.-b). *TC-Data.* Retrieved 2018-09-11, from `http://www.bitshapesoftware.com/instruments/tc-data/`

Bryan, N. J., Herrera, J., Oh, J., & Wang, G. (2010). MoMu : A Mobile Music Toolkit. *Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010)*(Nime), 174–177.

Cádiz, R., & Kendall, G. (2006). Fuzzy Logic Control Tool Kit : Real-time Fuzzy Control for Max / MSP and Pd. *Proceedings of International Computer Music ...* (Jaffe 1995), 341–346. Retrieved from `http://garykendall.net/papers/CadizKendall2006.pdf`

Cádiz, R. F., & Gonzalez-Inostroza, M. (2018). Fuzzy Logic Control Toolkit 2 . 0 : composing and synthesis by fuzzyfication. *NIME '18*.

Cádiz, R. F., & Sylleros, A. (2017). Arcontinuo : the instrument of change. *NIME '17*, 499–500.

Cadoz, C., & Wanderley, M. M. (2000). Gesture-music. *Trends in gestural control of music*, 71–94. Retrieved from `http://www.music.mcgill.ca/{~}mwanderley/Trends/Trends{_}in{_}Gestural{_}Control{_}of{_}Music/DOS/P.CadWan.pdf`

Camurri, A., De Poli, G., Leman, M., & Volpe, G. (2001). A Multi-layered Conceptual Framework for Expressive Gesture Applications. *In Proceedings of MOSART: Workshop on Current Directions in Computer Music*(1), 29–34.

Camurri, A., Volpe, G., De Poli, G., & Leman, M. (2005). Communicating expressiveness and affect in multimodal interactive systems. *IEEE Multimedia*, *12*(1), 43–53.

Caramiaux, B., Françoise, J., Schnell, N., & Bevilacqua, F. (2014). Mapping Through Listening. *Computer Music Journal*, *38*(3), 34–48. Retrieved from `http://www.mitpressjournals.org/doi/10.1162/COMJ{_}a{_}00255`

Caramiaux, B., & Tanaka, A. (2013). Machine Learning of Musical Gestures. *Proceedings of the International Conference on New Interfaces for Musical Expression 2013 (NIME 2013)*, 513–518. Retrieved from `http://nime2013.kaist.ac.kr/`

chen You, H., & Chen, K. (2007). Applications of affordance and semantics in product design. *Design Studies*, *28*(1), 23–38.

Cox, E. (1994). *The fuzzy systems handbook : a practitioner's guide to building and maintaining fuzzy systems*. Boston: AP Professional.

Crevoisier, A., Bornand, C., Guichard, A., Matsumura, S., & Arakawa, C. (2006). Sound Rose : Creating Music and Images with a Touch Table. *NIME '06 Proceedings of the 2006 conference on New interfaces for musical expression*, *1*, 212–215. Retrieved from `http://www.nime.org/proceedings/2006/nime2006{_}212.pdf`

Cycling '74. (n.d.). *Max Software Tools for Media.* Retrieved 2018-09-21, from `https://cycling74.com/products/max`

Dahl, S., Bevilacqua, F., Bresin, R., Clayton, M., Leante, L., Poggi, I., & Rasamimana, N. (2010). Gestures in Performance. In *Musical gestures: Sound, movement, and meaning* (pp. 36–68).

De Poli, G. (2004). Methodologies for Expressiveness Modelling of and for Music Performance. *Journal of New Music Research*, *33*(3), 189–202. Retrieved from http://www.tandfonline.com/doi/abs/10.1080/0929821042000317796

Demicco, R., & Klir, G. J. (2004). *Fuzzy logic in geology*. Amsterdam, Boston: Elsevier Academic Press.

Dimitrov, V., & Hodge, B. (2002). *Social fuzziology. study of fuzziness of social complexity*. Heidelberg,New York: Physica-Verlag.

Dualo. (n.d.). *du-touch.* Retrieved 2017-03-31, from https://dualo.org/du-touch/

Fabiani, M., Bresin, R., & Dubus, G. (2012). Interactive sonification of expressive hand gestures on a handheld device. *Journal on Multimodal User Interfaces*, *6*(1-2), 49–57.

Fels, S., Gadd, A., & Mulder, A. (2002). Mapping transparency through metaphor: towards more expressive musical instruments. *Organised Sound*, *7*(02), 109–126. Retrieved from http://www.journals.cambridge.org/abstract{_}S1355771802002042

Fenza, D., Mion, L., Canazza, S., & Rodà, A. (2005). Physical movement and musical gestures: A multilevel mapping strategy. *Proceedings of Sound and Music Computing Conference, SMC 2005*, 27. Retrieved from http://www.scopus.com/inward/record.url?eid=2-s2.0-84905187477{&}partnerID=40{&}md5=8ecf57292b9c0a47f55b32bfb4acc4a4

Fiebrink, R., & Caramiaux, B. (2016). The Machine Learning Algorithm as Creative Musical Tool. In *Oxford handbook on algorithmic music.* Retrieved from http://arxiv.org/abs/1611.00379

Geiger, G. (2006). Using the Touch Screen as a Controller for Portable Computer Music Instruments. *NIME '06 Proceedings of the 2006 conference on New interfaces for musical expression*, 61–64. Retrieved from http://www.nime.org/proceedings/2006/nime2006{_}061.pdf

Gillian, N., & Paradiso, J. A. (2012). Digito : A Fine-Grain Gesturally Controlled Virtual Musical Instrument. *NIME 2012 Proceedings of the International Conference on New Interfaces for Musical Expression*, 39–42.

Gonzalez-Inostroza, M., Sylleros, A., & Cádiz, R. (2017). Understanding interaction in musical multi-touch surfaces. In *2017 icmc/emw - 43rd international computer music conference and the 6th international electronic music week.*

Hachimura, K., Takashina, K., & Yoshimura, M. (2005). Analysis and evaluation of dancing movement based on LMA. *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, *2005*, 294–299.

Haken, L., Tellman, E., & Wolfe, P. (1998). An Indiscrete Music Keyboard. *Computer Music Journal*, *22*(1), 30–48. Retrieved from http://www.jstor.org/stable/10.2307/3681043

Hexler. (n.d.). *TouchOSC.* Retrieved 2017-04-14, from https://hexler.net/software/touchosc

Holland, S., Wilkie, K., Mulholland, P., & Seago, A. (2013). Music Interaction: Understanding Music and Human-Computer Interaction. In *Music and human-computer interaction* (pp. 1–28). Retrieved from http://link.springer.com/10.1007/978-1-4471-2990-5

Huang, S.-l., Hung, S.-y., & Chen, C.-p. (2015). Clustering-Based Multi-Touch Algorithm Framework for the Tracking Problem with a Large Number of Points. *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, 719–724.

Hunt, A., Wanderley, M. M., & Paradis, M. (2003). The Importance of Parameter Mapping in Electronic Instrument Design. *Journal of New Music Research*, *32*(4), 429–440. Retrieved from http://portal.acm.org/citation.cfm?id=1085207{%}5Cnhttp://www.nime.org/2004/NIME02/hunt.pdf

JazzMutant. (2017). *Lemur, multitouch modular controller for sequencers, synthesizers, virtual instruments, vj and light.* Retrieved 2017-04-12, from http://www.jazzmutant.com/lemur{_}overview.php

Jensenius, A. R., Wanderley, M. M., Godøy, R. I., & Leman, M. (2010). Musical Gestures. *Musical Gestures: Sound, Movement, and Meaning*, 12. Retrieved from http://arj.no/portfolio/publications/Jensenius{_}2010e.pdf

Johnston, B., Vallis, O., & Kapur, A. (2012). A Comparative User Study of Two Methods of Control on a Multi-Touch Surface for Musical Expression. *NIME 2012 Proceedings of the International Conference on New Interfaces for Musical Expression*, 370–373.

Johnston, C. T., & Bailey, D. G. (2008). FPGA implementation of a single pass connected components algorithm. *Proceedings - 4th IEEE International Symposium on Electronic Design, Test and Applications, DELTA 2008*, 228–231.

Jordá, S., Julià, C. F., & Gallardo, D. (2010). Interactive surfaces and tangibles. *XRDS: Crossroads, The ACM Magazine for Students*, *16*(4), 21.

Karg, M., Samadani, A. A., Gorbet, R., Kuhnlenz, K., Hoey, J., & Kulic, D. (2013). Body movements for affective expression: A survey of automatic recognition and generation. *IEEE Transactions on Affective Computing*, *4*(4), 341–359.

Kellum, G., & Crevoisier, A. (2009). A Flexible Mapping Editor for Multi-touch Musical Instruments. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 242–245. Retrieved from http://future-instruments.net/resources/Publications/GKellum{_}NIME09.pdf

Klir, G. J., & Yuan, B. (1995). *Fuzzy sets and fuzzy logic : theory and applications*. Upper Saddle River N J: Prentice Hall PTR.

Kosko, B. (1993). *Fuzzy thinking. the new science of fuzzy logic*. New York: Hyperion.

Kreković, G., & Pošćić, A. (2015). Shaping Microsound Using Physical Gestures. *Conference on Computation, Communication, Aesthetics and X*.

Kvifte, T. (2008). On the Description of Mapping Structures. *Journal of New Music Research*, *37*(4), 353–362.

Laban, R., & Ullmann, L. (1975). *The Mastery of Movement*.

Laiyang, L., & Junjun, G. (2014). Fingers' movement analysis based on Labanotation. *Advanced Research and Technology in ...*, 1307–1311. Retrieved from `http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6976523`

Landy, L. (2001). From algorithmic jukeboxes to zero-time synthesis: a potential A-Z of music in tomorrow's world (a conference provocation). *Organised Sound*, *6*(2), 91-96.

Larboulette, C., & Gibet, S. (2015). A Review of Computable Expressive Descriptors of Human Motion. *Proceedings of the 2nd International Workshop on Movement and Computing - MOCO '15*, 21–28. Retrieved from `http://dl.acm.org/citation.cfm?id=2790994.2790998`

Leman, M., Camurri, A., Loke, L., Larssen, A. T., Robertson, T., & Edwards, J. (2005). Understanding musical expressiveness using interactive multimedia platforms. *Musicae Scientiae*, *10*(1), 209–233. Retrieved from `http://journals.sagepub.com/doi/10.1177/102986490601000110`

Leman, M., & Maes, P.-J. (2014). The role of embodiment in the perception of music. *Empirical Musicology Review*, *9*(3-4), 236–246.

Linn, R., & Roger Linn Design. (n.d.). *LinnStrument.* Retrieved 2017-04-12, from http://www.rogerlinndesign.com/linnstrument.html

Loke, L., Larssen, A. T., Robertson, T., & Edwards, J. (2007). Understanding movement for interaction design: Frameworks and approaches. *Personal and Ubiquitous Computing*, *11*(8), 691–701.

Madronalabs. (n.d.). *Soundplane.* Retrieved 2017-04-12, from http://madronalabs.com/soundplane

Maes, P. J., Leman, M., Lesaffre, M., Demey, M., & Moelants, D. (2010). From expressive gesture to sound: The development of an embodied mapping trajectory inside a musical interface. *Journal on Multimodal User Interfaces*, *3*(1), 67–78.

Maes, P.-J., Leman, M., Palmer, C., & Wanderley, M. M. (2014). Action-based effects on music perception. *Frontiers in psychology*, *4*(January), 1008. Retrieved from http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3879531{&}tool=pmcentrez{&}rendertype=abstract{%}5Cnhttp://journal.frontiersin.org/article/10.3389/fpsyg.2013.01008/abstract

Magnusson, T. (2010). Designing constraints: composing and performing with digital musical systems. *Computer Music Journal*, *34*(4), 62–74. Retrieved from http://www.mitpressjournals.org/doi/abs/10.1162/COMJ{_}a{_}00026

Mancini, M., Varni, G., Kleimola, J., Volpe, G., & Camurri, A. (2010). Human movement expressivity for mobile active music listening. *Journal on Multimodal User Interfaces*, *4*(1), 27–35.

Marshall, M. T. (2009). *Physical Interface Design for Digital Musical Instruments* (Unpublished doctoral dissertation).

Mcglynn, P., Lazzarini, V., Depal, G., & Chen, X. (2012). Recontextualizing the Multi-touch Surface. *NIME 2012 Proceedings of the International Conference on New Interfaces for Musical Expression*, 459–463.

McNeill, D., & Freiberger, P. (1993). *Fuzzy logic*. New York: Simon & Schuster.

Mcpherson, A., & Kim, Y. (2011). Design and Applications of a Multi-Touch Musical Keyboard. *Proc. SMC*, *31*(1-3), 455–526. Retrieved from http://psycnet.apa.org/psycinfo/1927-01871-001

Misa Digital. (n.d.). *misa kitara.* Retrieved 2017-04-14, from https://misa-digital.myshopify.com/products/kitara

Mooney, J. (2010). Frameworks and affordances: Understanding the tools of music-making. *Journal of Music, Technology and Education*, *3*(2), 141–154.

Norman, D. A. (2013). *The Design of Everyday Things* (Vol. 16) (No. 4). Retrieved from http://ucdwiki.chuank.com/uploads/Main/UCDReading{_}wk5.pdf

Paine, G. (2009). Towards unified design guidelines for new interfaces for musical expression. *Organised Sound*, *14*(02), 142. Retrieved from http://link.springer.com/chapter/10.1007/978-3-319-07668-3{_}62{#}page-1

Paine, G. (2013). New Musical Instrument Design Considerations. *MultiMedia, IEEE*, *20*(4), 76–84. Retrieved from http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6674037{%}5Cnhttp://www.scopus.com/inward/record.url?eid=2-s2.0-84890091333{&}partnerID=40{&}md5=d402c0bd22c9fff877afce5b0451e0b9{%}5Cnhttp://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6674037{%}5Cnhttp://www.

Paradiso, J. A. (2014). The Gesture Recognition Toolkit. *Journal of Machine Learning Research*, *15*, 3483–3487.

Paradiso, J. A., & O'Modhrain, S. (2003). Current Trends in Electronic Music Interfaces . Guest Editors ' Introduction. *Journal of New Music Research*, *32*(4), 345–349.

Pd Community. (n.d.). *Pure Data.* Retrieved 2018-09-21, from https://puredata.info/

Pelachaud, C. (2009). Studies on gesture expressivity for a virtual agent. *Speech Communication*, *51*(7), 630–639. Retrieved from http://dx.doi.org/10.1016/j.specom.2008.04.009

Perrotin, O., & D'Alessandro, C. (2013). Adaptive mapping for improved pitch accuracy on touch user interfaces. *Proceedings of the International Conference on New Interfaces for Musical Expression*, *91403*, 186–189. Retrieved from http://nime2013.kaist.ac.kr/

Rekimoto, J., & Schwesig, C. (2006). PreSenseII: bi-directional touch and pressure sensing interactions with tactile feedback. *CHI '06 Extended Abstracts on Human Factors in Computing Systems (CHI EA '06).*, 1253–1258. Retrieved from http://dl.acm.org/citation.cfm?id=1125685

Ren, Z., Mehra, R., Coposky, J., & Lin, M. (2012). Designing virtual instruments with touch-enabled interface. *Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts*, 433–436. Retrieved from http://doi.acm.org/10.1145/2212776.2212820

Roads, C., & Strawn, J. (1996). *The Computer Music Tutorial*. MIT Press. Retrieved from https://books.google.cl/books?id=nZ-TetwzVcIC{&}pg=PA239{&}dq=double+carrier+fm{&}hl=es-419{&}sa=X{&}ved=0ahUKEwjY{_}fK0ubndAhUHDJAKHeW1AiQQ6AEIKDAA{#}v=onepage{&}q=doublecarrierfm{&}f=false

Roberts, C., Wright, M., Kuchera-Morin, J., & Höllerer, T. (2014). Rapid Creation and Publication of Digital Musical Instruments. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 239–242. Retrieved from `http://www.nime.org/proceedings/2014/nime2014{_}373.pdf`

Roli. (2018). *ROLI — Seaboard RISE.* Retrieved 2017-03-30, from `https://roli.com/products/seaboard-rise/5d-touch`

Salazar, S. (2014). miniAudicle for iPad Touchscreen-based Music Software Programming. *ICMC*, 686–691.

Samadani, A. A., Burton, S., Gorbet, R., & Kulić, D. (2013). Laban effort and shape analysis of affective hand and arm movements. *Proceedings - 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, ACII 2013*, 343–348.

Samadani, A. A., Dehart, B. J., Robinson, K., Kulic, D., Kubica, E., & Gorbet, R. (2011). A study of human performance in recognizing expressive hand movements. *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, 93–100.

Sarwate, A., & Snyder, J. (2014). SkipStep : A Multi-Paradigm Touch-screen Instrument. *ICMC 2014*(September), 748–753.

Schlei, K. (2012). TC-11 : A Programmable Multi-Touch Synthesizer for the iPad. *NIME 2012 Proceedings of the International Conference on New Interfaces for Musical Expression*, 288–291.

Schlei, K. (2015). *TC-11 User Guide Version 2 . 0* (Tech. Rep.).

Schöning, J., Brandl, P., Daiber, F., Echtler, F., Hilliges, O., Hook, J., … Zadow, U. V. (2008). *Multi-Touch Surfaces : A Technical Guide Technical Report TUM-I0833 Categories and Subject Descriptors* (Tech. Rep.). Retrieved from `http://www.mendeley.com/research/multitouch-surfaces-a-technical-guide/`

Schumacher, M., Wanderley, M. M., Schumacher, M., & Wanderley, M. M. (2016). Integrating gesture data in computer-aided composition : A framework for representation , processing and mapping. *Journal of New Music Research*, *8215*(January), 0–14. Retrieved from http://dx.doi.org/10.1080/09298215.2016.1254662

Street, E., Volioti, C., Katsouli, E., & Manitsaris, A. (2016). x2Gesture : how machines could learn expressive gesture variations of expert musicians. *NIME '16 Proceedings of the 2016 conference on New interfaces for musical expression*(Section 5).

Suda, N., & Vallis, O. (2014). Touchpoint: Dynamically re-routable effects Processing as a multi-touch tablet instrument. *Proceedings - 40th International Computer Music Conference, ICMC 2014 and 11th Sound and Music Computing Conference, SMC 2014 - Music Technology Meets Philosophy: From Digital Echos to Virtual Ethos*(September), 1469–1476.

Sylleros, A., de la Cuadra, P., & Cádiz, R. F. (2014). Designing a Musical Instrument: Enlivening Theory Through Practice-Based Research. *Design Issues*, *30*(2), 83–96.

Sylleros, A., de la Cuadra, P., & Cádiz, R. F. (2017). Understanding the Quality of Subject–Object Interaction: A Disciplinary Model for Design Validation. *The Design Journal*, *20*(1), 67–86. Retrieved from https://www.tandfonline.com/doi/full/10.1080/14606925.2017.1252542

Touch Innovations. (2017). *Elite.* Retrieved 2017-04-12, from https://touchinnovations.com/products/elite/

Trump, S., & Bullock, J. (2014). Orphion: A Gestural Multi-Touch Instrument for the iPad. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 159–162. Retrieved from http://www.nime.org/proceedings/2014/nime2014{_}277.pdf

Truong, A., Boujut, H., & Zaharia, T. (2016). Laban descriptors for gesture recognition and emotional analysis. *Visual Computer*, *32*(1), 83–98. Retrieved from http://dx.doi.org/10.1007/s00371-014-1057-8

Visi, F., Coorevits, E., Schramm, R., & Miranda, E. R. (2017). Musical Instruments, Body Movement, Space, and Motion Data: Music as an Emergent Multimodal Choreography. *Human Technology*, *13*(1), 58–81. Retrieved from http://humantechnology.jyu.fi/archive/vol-13/issue-1/musical-instruments-body-movement-space-and-motion-data/@@display-file/fullPaper/Visi{_}Coorevits{_}Schramm{_}Reck-Miranda.pdf

Visi, F., Schramm, R., & Miranda, E. (2014). Gesture in performance with traditional musical instruments and electronics - Use of embodied music cognition and multimodal motion capture to design gestural mapping strategies. *Proceedings of the International Workshop on Movement and Computing*, 100–105.

Volpe, G. (2003). *Computational models of expressive gesture in multimedia systems* (Unpublished doctoral dissertation).

Von Altrock, C. (1997). *Fuzzy logic and neurofuzzy applications in business and finance*. Upper Saddle River, New York: Prentice-Hall.

Walbeck, J., Snow, R., & Ph, D. (2013). CAPACITANCE SENSING GESTURAL MUSIC CONTROLLER Music Science and Technology Program Department of Music Technology Program. , 457–461.

Wanderley, M. M., & Depalle, P. (2004). Gestural Control of Sound Synthesis. *Proceeding of the IEEE*, *92*(4).

Wanderley, M. M., & Orio, N. (2002). Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI. *Computer Music Journal*, *26*(3), 62–76.

Wang, G. (2011). Designing for the iPad : Magic Fiddle. *Proceedings of the International Conference on New Interfaces for Musical Expression*(June), 197–202.

Wessel, D., & Wright, M. (2002). Problems and Prospects for Intimate Musical Control of Computers. *Computer Music journal*, *26*(3), 11–14. Retrieved from `http://www.nime.org/proceedings/2001/nime2001{_}011.pdf`

Wickens, C. D., Gordon, S. E., & Liu, Y. (1998). Human-Computer Interaction. In *An introduction to human factors engineering* (pp. 451–492).

Wizdom Music. (n.d.). *GeoShred.* Retrieved 2018-09-20, from `http://www.wizdommusic.com/products/geoshred.html`

Yen, J., & Langari, R. (1999). *Fuzzy logic : intelligence, control, and information*. Upper Saddle River, N.J.: Prentice Hall.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*, 338-353.