Cephalomorphic interface for emotion-based musical synthesis

report

Vassilios-Fivos A. Maniatakos
LIMSI, CNRS

Master ATIAM, University Paris VI & IRCAM

Acknowledgments

Many thanks to my thesis supervisor, professor Christian Jacquemin for giving me the opportunity to work on such a wonderful project. I am grateful for the interest he was showing for me throughout my internship at LIMSI, as well as for his invaluable scientific advice and help.

I would also like to thank all the AMI team of LIMSI laboratory, for the perfect collaboration. Thanks to Ramy Ajaj, who was always there for me when I needed. Special thanks to Arshia Cont, I enjoyed the discussion we had for Neural Networks.

I would also like to thank Elissabet for her patience and understanding for my ‘particular’ 6month relationship with Pogany...

For the end, I would like to thank my family for their continuous support. This work is dedicated to them.

Contents

1 Introduction **1**

1.1 Human-Computer Interaction . . . 1

1.1.1 The study of emotions . . . 1

1.1.2 Gesture for HCI . . . 1

1.1.3 Gesture's importance for music computing and music interface design strategies 2

1.2 Scope, context, structure of the report . . . 2

2 bibliography **4**

2.1 gesture interfaces . . . 4

2.2 an overview on different types of applications that employ gesture recognition . . . 5

2.3 gesture interfaces for music . . . 5

2.4 Expressive multimodal interfaces for music . . . 7

2.4.1 approaches . . . 8

2.4.2 Proposed models for evaluation of expressive interfaces . . . 9

3 gesture recognition strategies **10**

3.1 Gesture interfaces and methods used for gesture recognition . . . 10

3.2 HMMs . . . 11

4 The model **15**

4.1 Pogany: An affective facial interface . . . 15

4.2 Description of the interface . . . 15

4.3 Overview of Pogany as an expressive music interface . . . 17

4.4 Front part . . . 17

5 Middle part **19**

5.1 Gesture Analysis-feature extraction . . . 19

5.2 Gesture Recognition module . . . 22

5.2.1 HMM machine . . . 22

5.2.2 Experiments for gesture recognition and strategies for HMM configuration . . 23

5.2.3 results . . . 24

5.2.4 Conclusions for isolated gesture recognition . . . 25

5.3 Real-time recognition for Pogany . . . 25

5.3.1 Overview . . . 25

5.3.2 gesture segmentation . . . 26

5.3.3 HMM configuration . . . 27

5.3.4 Training 27
5.3.5 Real-time Recognition: Viterbi decoding 30
5.3.6 Implementations for off-line training and real-time recognition 32

6 Mapping and music synthesis strategies 35
6.1 An overview on strategies for effective interaction 35
6.1.1 direct mapping strategies 35
6.1.2 indirect mapping strategies 36
6.2 Synthesis methods: The background 36
6.2.1 FM synthesis theory and attributes 36
6.2.2 Granular synthesis 37
6.3 mapping strategies implemented for Pogany 37
6.3.1 Direct Mapping strategies 37
6.3.2 Indirect mapping strategies 40
6.3.3 Environment 41

7 Evaluation of 'Pogany' as an interface for sound interaction 43
7.1 Context 43
7.2 Preparation of the experiment 44
7.2.1 System configuration 44
7.2.2 Conditions of the experiment 45
7.3 the experiment 45
7.3.1 results 46

8 Conclusions-Future research 50
8.1 General conclusions 50
8.2 Future Work 51

Bibliography 53

Abstract

The subject of this master thesis is to adapt ”Pogany”, a tangible cephalomorphic interface designed and realized in LIMSI-CNRS laboratory, Orsay-France, to use for music synthesis purposes. We are interested in methods for building an affective emotion-based system for gesture identification, captured through a facial interface that understands variations of luminosity by distance or touch. After a brief discussion on related research, the report introduces issues in the conceptualization and development of a gesture learning and real-time recognition tool based on HMM theory and technology. Employing direct features and mapping techniques, but also giving priority to the importance of high-level information arising from user's gesture, we have built a system for expressive music performance. In this report, gesture training and recognition system used in this scope are thoroughly presented and evaluated. In order to evaluate the interface's overall performance as a music expressive medium, we then describe an experiment with several subjects interacting musically with Pogany and discuss our results. Finally, we give clues for explicit future work, as well as further possible improvements concerning the interface, the recognition system and its mapping with a music synthesis tool.

Chapter 1

Introduction

1.1 Human-Computer Interaction

Human - computer interaction(HCI) plays a role of growing importance as computer technology continues to evolve. Trends towards embedded, ubiquitous computing in domestic environments creates the necessity to try for something more than the traditional desktop paradigm: keyboard, mouse and display should be considered as insufficient for what is described in [1] as *everyday computing.* Instead, natural actions in human-to-human communication, such as speak and gesture, have been a source of inspiration in an effort to create natural, convenient and efficient computer interfaces [2]. Thus, for HCI, the goal of gesture interpretation was primarily set as to push the envelope of advanced human-machine communication to bring the performance of human-machine interaction closer to human-human interaction [3].

1.1.1 The study of emotions

Research on natural communication between physical and virtual posed some new priorities for HCI. Towards this direction, HCI research field finds it useful to borrow models, theories and methodology from psychology, social sciences and art, keeping with them a bidirectional relationship. Furthermore, as described in [4]: *if for many years research was devoted to the investigation of more cognitive aspects, in the last ten years lot of studies emerged on emotional processes and social interaction.* Consequently, during the last decade HCI has focussed on two very important subcategories of what we generally mean by the word 'interfaces': The *multimodal* and the *affective* interfaces. The first describes interfaces which employ information coming from several channels in order to build an application focused and on the user and constrained by his/her need for ergonomic interaction [4]. The second refers to a user interface that appeals to the emotional state of users and allows to express themselves emotionally, as well as to receive information with emotional content.

1.1.2 Gesture for HCI

The shift of interest on natural emotion-based interaction suggests gesture interpretation as a very important field of research for Human Computer Interaction. Gesture, defined as *a movement of the body that contains information* [5], or as *expressive meaninful body motions with the intent to convey information or interact with the environment* [6] is regarded as a powerful carrier of emotional content and main channel of non-verbal communication. Additionally, new motion capture technologies and cost-effective powerful personal computers further encourage HCI to adopt gesture recognition methodology. In this framework, as a mean to form implicit messages of high-level emotional content, expressive gesture plays significant role in the development of innovative multimodal interactive systems able to provide users with natural expressive interfaces [7].

1.1.3 Gesture's importance for music computing and music interface design strategies

Under these conditions gesture, not only as a part of art performance (i.e definite gesture of fingers playing on a music instrument) but also as a statement of an emotional content to transform to artistic expression was not far to come. The first connection between unfettered gesture and electronic music control was firstly introduced by Theremin in the 1920s. Concepts for gestural control of music have also benefited from the cross-fertilization with other expanding research fields and applications, such as HCI, human gesture recognition, computer animation, and biomechanics, to name a few,as stated in [8]. Since then many scientists and artists found this idea inspiring and try to develop such interdisciplinary systems for both scientific and artistic purposes. Nowadays, control of sound/music based on movements is still considered as an outstanding challenge especially when treated in a more sophisticated basis than before. Such a basis, as far as multimodal affective interfaces are concerned, could be considered as:

- building the framework to *identify, describe* and *represent* an expressive gesture as a high-level information from a collection of cues,
- conceptualizing *the strategies* for expressive musical interaction, in other words to put the user into an effective dialog between her/him and the multimodal interface.

1.2 Scope, context, structure of the report

We argue that high-level gestural information can be revealing for the expressive diathesis or emotional state of the user. In this scope, an interface that succeeds in decoding such information can be very useful to use for the control of a virtual music instrument. Using strategies 1) to decode high-level information from gesture 2) to create the link between this information and its semantic meaning 3) to create intelligent correspondences between these semantics and other direct gestural analysis parameters to music parameters , we have built and evaluated a music interaction system for the 'Pogany' interface.

The whole work of this master thesis was mightily bound with the 'Pogany' interface, an interface conceptualized and constructed in LIMSI laboratory at Orsay-France. 'Pogany' has the form of a stylized human head in the size of a joystick, and detects gesture by distance or touch. Its anthropomorphic type has particular meaning for our research, as it permits the link of the gestures to recognize with the face emotions; the last will be further discussed in several sections of the report.

Following this chapter, chapter 2 introduces the research field of gestures interfaces. After and overview over gesture recognition applications, we focus on the gesture interfaces for music and especially to expressive multimodal interfaces. In chapter 3 we provide a global view over the gesture recognition methods employed with interfaces, giving priority to the Hidden Markov Models (HMM) approach and presenting their basic principles. In Chapter 4 we describe the interface and provide the reader with a general image of the music interaction system build. The front part of the system is described at the end of the chapter, while in chapter 5 we analyze our strategies for feature extraction and HMM off-line and real-time recognition. In chapter 6 we discuss our mapping and music synthesis strategies used. In 7 we provide the reader with the experimental processes followed in order to evaluate ‘Pogany’ as an interface for sound interaction. Finally, in chapter 8 we discuss the issues arising from this research, reaching a number of global conclusions and giving clues for future research plans.

Chapter 2

bibliography

In this chapter, we will first provide a short description of what is a gesture interface, its scope and purpose. After an overview on the range and kind of applications that employ gesture recognition thechniques, a short review of the music interfaces research field and issues concerning their conceptualization, construction and evaluation will serve as an introduction to the aspects of the area of our interest: the expressive music interfaces.

2.1 gesture interfaces

In a rather simplified view, a *gesture interface* is an interface where users specify commands by simple gestures such as drawings and actions [58]. The key issues in its design are how to capture gesture information and how to recognize the gestures from captured data. To develop a gesture interface, we need some criteria to *evaluate its performance*, such as: meaningful *gestures*; suitable *sensors*; efficient training *algorithms*; and accurate, efficient, *on-line/real-time recognition.*

Concerning the last, recognition of human gestures is a research area with increasing importance for multimodal human computer interfaces [9]. Different approaches have been made to this direction; in modern times, such a task precludes the use of computer systems. In a first view, the technology for capturing gestures is expensive; this means that additional components-when compared with a simple computer configuration- such as sensors and sensor communication devices, a vision system or a dataglove are often needed. For this reason some graphical devices, such as a mouse, light pen, joystick, trackball, touch tablet, and thumb-wheel, can be employed to provide a simple input to a gesture recognizer. Other possible devices are a foot controller, knee controller, eye tracker, data nose, and tongue-activated joystick.

Apart from sensor-based techniques, the use of which was spread the previous decades, the last years there is a growing interest for vision-based gesture recognition systems. This is partly due to the decrease of costs when using video technology, the creation of protocols driving digital communication between camera equipment and computers, as well as the achievements of research on off-line / real-time image recognition.

Despite the variety of the types of gesture interfaces, there are some common principles among all such devices, which have to do with the kind of processing that must be able to provide. Hunt and Kirk in [21] consider various attributes as characteristics of a real-time multiparametric control systems. Some of these are:

- There is no fixed ordering to the human-computer dialogue.

- There is no single permitted set of options (e.g. choices from a menu) but rather a series of continuous controls.
- There is an instant response to the users movements.
- The control mechanism is a physical and multi-parametric device which must be learned by the user until the actions become automatic.
- Further practice develops increased control intimacy and thus competence of operation.
- The human operator, once familiar with the system, is free to perform other cognitive activities whilst operating the system (e.g. talking while driving a car).

In the following, after a short overview of different kind of applications based on gesture recognition, we will discuss how the previous principles apply to the music synthesis domain.

2.2 an overview on different types of applications that employ gesture recognition

The multiple domains of interest for gesture recognition have resulted to an enormous range of applications. Such application systems cover domains such as virtual environments, smart surveillance, robotics, sign language translation, etc. Concerning the first, conventional use of hands in most of the human activities has lead research on virtual environments to use this metaphor and to the employ hand gesture recognition techniques in order to bring difficult-to-define tasks down to a point of realizable interactivity. For instance, in 1997 Zeller et al. [11] presents a visual environment for very large scale biomolecular modeling application. This system permits interactive modeling of biopolymers by linking a 3D molecular graphics and molecular dynamics simulation program. Hand gestures serve as the input and controlling device of the virtual environment. Concerning surveillance, an example of possible applications is the one by Quek[15]: he presents a FingerMouse application to recognize 2-D finger movements which are the input to the desktop. Of course, in the area of surveillance there is a glance of private research for security companies or police and military purposes. In robotics, Triesch and Maslburg[17] develop a person-independent gesture interface on a real robot which allows the user to give simple commands such as how to grasp an object and where to put it. As far as sign language translation systems are concerned, Imagawa et al.[20] implement a bi-directional translation system between Japanese Sign Language (JSL) and Japanese in order to help the hearing impaired communicate with normal speaking people through sign language.

It is worth to mention that this variety of applications permitted a multi-variate approach in terms of the criteria according to which the classification is made. For instance, it sounds reasonable that a module for sign language recognition should differ in separability factors from a system targeted to recognize gesture in a more general scope. Top-down trends to the gesture recognition problems discuss issues such as elastic deformation of schemata such as in [18]. In an approach to categorize gestures according to expressivity creteria, Camurri in [19] uses the theory of effort by Laban.

2.3 gesture interfaces for music

Gestural control of computer generated sound can be seen as a highly specialized branch of human-computer interaction (HCI) involving the simultaneous control of multiple parameters, timing, rhythm, and user training. According to Hunt and Kirk:

In stark contrast to the commonly accepted choice-based nature of many computer interfaces are the control interfaces for musical instruments and vehicles, where the human operator is totally in charge of the action. Many parameters are controlled simultaneously and the human operator has an overall view of what the system is doing. Feedback is gained not by on-screen prompts, but by experiencing the moment-by-moment effect of each action with the whole body.

Such characteristics of real-time multiparametric control as those mentioned in the previous section, help define the different contexts of interaction in music domain. According to [22], interaction in a musical context may mean:

- *instrument manipulation* (performer-instrument interaction) in the context of real-time sound synthesis control.
- *device manipulation in the context of score-level control*, e.g. a conductors baton used for indicating the rhythm to a previously defined computer generated sequence
- *other interaction contexts related to traditional HCI interaction styles*, such as drag and drop, scrubbing or navigation.
- *device manipulation in the context of post-production activities*, for instance in the case of gestural control of digital audio effects or sound spatialisation.
- *interaction in the context of interactive multimedia installations* (where one person or many peoples actions are sensed in order to provide input values for an audio/visual/haptic system).

The first devices used referred mainly to the musician-instrument scheme of interaction. In the case, the instrument is an electronic or computer synthesis module. In [23], one can find a reference guide for electronic music interfaces, including switches, potentiometers, motorized faders, joysticks and trackballs, digitized tablets and touch sensitive tabs, touch sensitive screens, various keyboards setups and extensions (i.e three-dimensional keyboard) ribbon controllers and breath controllers: these controllers were and are still used for music interaction purposes. From the time of analog electronic instruments, and the cables and knobs, newer controllers aim to enhance interactivity, speed of response, multiple level control, and simultaneous control of different music parameters.

However, the boundaries of the interaction schemes in which a music interface can be employed are rather indefinite, as different configurations for the same interface can result to different applications. Therefore, if it is for dividing music gesture interfaces in categories, a classification according to the type of the input could prove helpful. A first approach would reveal two big categories: sensor and non-sensor interfaces.

Concerning the first, we could classify them in two large subcategories: those which employ spatial and those with body sensors. Well-known past examples of spatial-sensor interfaces include the Radio Drum (Mathews), which measures the location of two batons in three dimensions in relation to a rectangular radio receiver and Donald Buchla's Lightning[24], which uses an infrared signal to locate a performer within a user-definable two-dimensional grid. Examples of body sensors include MIDI Dancer [28], which analyzes body shape by measuring the angles of arm, leg, and hip joints.

The second large category includes the interfaces that also use other input devices. The last years, music interfaces were inspired by computer vision techniques, which permit to analyze gesture by a video stream in real time. The latest has lead to a variety of software based gesture interfaces for music control. The enhancements of capture techniques in terms of multimodality has encouraged the conceptualization of systems using alternative capture systems. To name a few, in 2002, in [33] the authors present The Termenova, an interface close to the concept of Theremin, which offers indeed enhanced training capabilities and a perceivable structure for the performer (adjustable laser module). In 2005, in the PHASE project[34], gesture and haption, as well as metaphors that lead interaction procedure with sound an music was extensively studied, and led to a multi-modal demonstrator intended for the general public. In 2006, a new interface to drive speech and song performance (GRASP[36]) were presented.

A particular subcategory concerns interfaces that employ multimodal technics in order to capture phycophysiological parameters. The first worth to mention results of such a research are BioMuse [29], which measures electrical voltages produced by muscle contraction, and a host of hand-based controllers such as The Hands[30], which also measures proximity. In their continuation, Tanaka and Knapp[31] in 2002 have applied the electromyogram to musical control. In 2006, Miranda and al. have presented a system employing a Brain-computer music interface to use for computer improvisation mapped to a piano, as an accompaniment to a 'real' piano player.

2.4 Expressive multimodal interfaces for music

Emotional expression plays a key role in musical expression [43]. However, expression in music is not something straightforward to define as a term. One reason for this is the different point of view for each field of interest (musicians, HCI researchers, music scientists, phsychologists). One of the common elements in these approaches lies on the view of expressivity as an independant channel of communication. Performers communicate musical expression to listeners by a process of coding. Listeners receive musical expression by decoding. Performers code expressive intentions using expressive- related cues (Brunswikian lens model). Extensive work has been done to identify most relevant cues. These cues include: tempo, sound level, timing, intonation, articulation, timbre, vibrato, tone attacks, tone decays and pauses.

From the part of music interfaces conceptualization methodology, it is though important to examine the phsycology approach in order to set up the basic rules for an interface, evaluate its attributes and ensure its expandability. In this light, in [43] the author provides a profound top-down approach to the expressivity phenomenon. The impact of expressivity in music as a multidimensional process is summarized to five prominent components: a) Generative rules that function to clarify the musical structure; (b) Emotional expression that serves to convey intended emotions to listeners; (c) Random variations that reflect human limitations with regard to internal time-keeper variance and motor delays; (d) Motion principles that prescribe that some aspects of the performance (e.g. timing) should be shaped in accordance with patterns of biological motion; and (e) Stylistic unexpectedness that involves local deviations from performance conventions.

The factors mentioned before tend to construct what we call expressivity in a performance and communicate information of such kind through a built multidimensional channel between performers and listeners. However, things get more difficult when trying to adapt such models to synthesis procedure, i.e when conceptualizing the parameters of an expressive interface. One of the problems lies on the immaturity of means for computer music: a common complaint about electronic and computer music is that it lacks expressivity. A lot of work has been done on this, with the development of complex algorithms for synthesis. Despite these efforts and the enhancements of computer synthesis sound, the source of the problem was discovered to be rather to the control of the synthesis parameters: just as in the acoustic instrument metaphor, where the musician applies expressivity through gesture. Thus, if one wishes to model expressivity, he/she rarely can avoid to model the gesture in all its complexity. However there is also a third point: just as important as the capture of the gesture itself is the manner in which the correspondance between gestural data and synthesis parameters is done, process that in terms of computer music is called *mapping*. For instance, this is what happens with additive synthesis: under careful modeling it can result to high quality synthetisized sound, however it is impossible to develop a system that gives control to the user to all this parameters simultaneously as well as a physical meaning to the control actions, in order for one to be able to make a successful mapping of these actions with the music parameters. Thus, apart from the case of one to one mapping, which may be the case for instance of physical modeling synthesis, synthesis based on signal models allows for higher level coupling between control gestures.

2.4.1 approaches

The last years, as research on expression and emotions in HCI has made important steps, there has been presented a number of interesting interfaces for expressive music interaction. The importance of these interfaces lies on their ability to establish an effective, non-trivial and sometimes a, affective interaction with their users. The Driving Interface for Expression Synthesis in the framework of the Expression Synthesis Project (ESP) presented in 2005 [44] uses *a compilling driving metaphor for expressive performance so as to make high-level expressive decisions accessible to non- experts.* With the moto: 'Almost everyone knows to drive a car, but not everyone can play an instrument', the authors have the ambition: 1) to make a computer synthesis system available to a large population of users 2) to minimize the learning curve for expressive performance. The driving metaphor performs a vast amount of similarities with an instrument; however, there are doubts that it can perfectly suit the music interaction phenomenon in expandability. Furthermore, the known-to-all driving patterns could set constraints for the explorability of the interface of the users: They are rather pushed more to certain types of manipulation than others, such as pressing the accelerator or brakes separately rather than simultaneously. However, the driving interface has been adopted in other real-time interactive music systems, such as Amplitude by 'Harmonixmusic' and the Harmonic Driving in Tod Machovers Brain Opera [45]. The same year, Yonezawa and al. [46], presented the HandySinger system, a hand-puppet interface to drive voice morphing in real time through the ESVM synthesizer.

Apart from such kind of interfaces, the approach through *augmented instruments* elects the expressivity approach in computer music. Augmented instruments are based on traditional instruments, where gestures are defined a priori, i.e in case of the violin, the different types of bow strokes from a widely accepted and formalized set of gestures. Different approaches are possible with 'augmented instruments'. First sensors can be utilized to add control possibilities that are not directly related to normal playing techniques. For instance, various buttons can be added to the body of instruments, which adds new gestures to the instrument player. Second, sensors can be applied to capture normal playing gestures. This approach consists of decoding high-level expressive information of a performer arising from conventional gesture or other components which can communicate information and be captured through multimodal techniques. In most of works, gesture data has been used as a control of sound filters, or as the input for physical synthesis. Fewer works have reported on interpreting gesture data, in order to provide high-level parameters used in the mapping design. One of the most important recent works in this direction is the augmented violin project in IRCAM. In [47], authors describe their intention to use the conventional gesture vocabulary in order to build an interpretation level from the data stream and facilitate the mapping between gestures and sounds.

In an approach to drag high-level information from the user, different gesture identification frameworks have been implemented based on various technologies. In 'augmented violin' project, high-level information from the violinist gesture is decoded through an HMM module. Other framework are based on Neural Network (N.N) technology for the same purpose. Cont and al. in [48] describes a Dynamic Network architecture to model such gestures, and in [49] he presents an implementation of a N.N based recognizer for the PureData real-time music framework. Recently, Modler and al. discusses a number of prominent features extracted from video analysis to feed to a neural network for the detection of animations of a real face.

2.4.2 Proposed models for evaluation of expressive interfaces

A lot of researchers address the topic of musical expression. Examples may be found in [39] or [40]. However, a measuring instrument to evaluate the specific potential for expressive music of interfaces is still under discussion.

Wanderley and Orio present a method based on tools used in research on Human Computer Interaction (HCI) to evaluate interfaces[41]. It includes evaluation of learnability, explorability, feature controllability and timing controllabillity. Learnability corresponds to the amount of difficulty when learning an interface. Explorability is the attribute that allows the user to enhance his technique on the instrument, i.e to discover the new sound capabilities for the instrument. The last two, feature and timing controlability, refer to the amount of control given to the user for the musical procedure. Most interface evaluation approaches are based on the one described by Wanderley. In [42],the author uses these tools and modifications to evaluate an accelerometer and a Korg Kaoss-pad two-dimensional controller. The evaluation method presented in this paper differs from previous work since measurement methodology is based on the expressive skills of musicians and the player estimation of the interface ability to deal with it.

Chapter 3

gesture recognition strategies

The recognition of human gestures is a research area with increasing importance for multimodal human computer interfaces [9]. In order for music synthesis, gesture interfaces with multimodal sensors are widely used to enhance expression capabilities. At this point, providing the interface with an additionnal module of decoding high-level gesture information is considered as of great importance, as it allows further types of processing and interaction. In the scope of our work, we implemented an Hidden Markov Model (HMM) application to control the gesture recognition procedure, as we have reasons to believe that in our situation HMM have several advantages other other pattern classification systems. In this chapter, we will make an introduction on the theory and aspects of the HMM gesture recognition methodology. The reasons for our approach and their result to the music synthesis system will be discussed in next chapters.

3.1 Gesture interfaces and methods used for gesture recognition

As mentioned in the introduction, there has been a variety of approaches to gesture recognition. One of the earliest approaches was the one described in [10] in 1979 based on dictionary lookup methods. Several methods have been used for gesture recognition: template-matching, dictionary lookup, statistical matching, linguistic matching, neural networks and ad hoc methods.

Statistical matching methods employ statistics of feature vectors to derive classifiers. Some of these methods make assumptions about the distributions of features within the class; that means to assume that features arising from gesture follow one of the known distributions. The performance of such systems is often poor as these assumptions are violated. Other statistical matching methods do not have such assumptions, but they require much training data to estimate the form of the distribution. The typical statistics used in such methods are average feature vector per class, per-class variance of the individual features, and per-class correlations within features.

Several statistical learning techniques have been used for gesture recognition, such as multidimensional discriminant analysis in [50] and low-level statistical features manipulation in [51], to name a few. In [52] eigenspace was introduced to represent an approximate value for gesture 3-D information . Some of the methods are suitable for only one type of feature representation, while others are more generally applicable.

3.2 HMMs

Due to the widespread popularity of Hidden Markov Models in speech recognition and handwriting recognition, HMMs have begun to be applied in spatio-temporal pattern recognition and computer vision. The HMM approach to gesture recognition is motivated by the successful application of Hidden Markov modeling techniques to speech recognition problems. The similarities between speech and gesture (Table 3.1) suggest that techniques effective for one problem may be effective for the other as well[58].

Table 3.1: speech vs gesture

speech	gesture
expressivity, high-level content	
vocabulary, structure	
phonemes	simple gesture
words	complex gestures
continuous sound signal	continuous signal 2-D, 3-D(image) or 1-D(sensors)
features extraction procedure	
MFCC,LPC	PCA
silence	inaction
time, place and social factors variance	

First, gestures, like spoken languages, vary according to location, time, and social factors. Second, body movements, like speech sounds, carry certain meanings. Third, regularities in gesture performances while speaking are similar to syntactic rules. Therefore, linguistic methods may be used in gesture recognition. On the other hand, gesture recognition has its own characteristics and problems. To develop a gesture interface, some criteria are needed to evaluate its performance such as meaningful gestures, suitable sensors, efficient training algorithms, and accurate, efficient, on-line/real-time recognition.

In general, the concept of HMM can be used in solving three basic problems: the evaluation problem, the decoding problem, and the learning problem. The reader can find a complete presentation in Rabiner's HMM tutorial [53]. An HMM 3.1 can be defined by:

- N - a set of states, including an initial state S_i and a final state S_f (sometimes states can also symbolized with ω). These states are hidden, but their importance lies onto their physical significance. Apart from the number N of states, interconnection between them plays a determinant role (figure 3.2).

- O- a set of observations $O_{1..k}$, $O_k \in V$, where V the set of possible emitting symbols (the vocabulary). The previous can take slightly different forms at more complex HMMs definitions, i.e for continuous or multidimensional HMMs.

- A - a transition probability matrix, A = $\{a_{i,j}\}_N$, where a_{ij} is the transition probability of taking the transition from state i to state j.

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], 1 \leq i, j \leq N \tag{3.1}$$

- B - The output probability matrix, $B = \{b_j(O_k)\}$ for discrete HMM's, $B = \{b_j(x)\}$ for a continuous HMM, where O_k stands for a discrete observation symbol, and x stands for continuous observations of k-dimensional random vectors.

$$b_j(k) = P[v_k(t)|q_t = S_j], 1 \leq j \leq N, 1 \leq k \leq M. \tag{3.2}$$

- $\{\pi_i\}$ -The initial state distribution, where $\pi = P[q_1 = S_i], 1 \leq i \leq N$. That means the possibility that the the first state of the sequence is the state S_i.

- λ- It refers to all the parameters that should be defined for an HMM. Given N, we have $\lambda = (A, B, \pi)$.

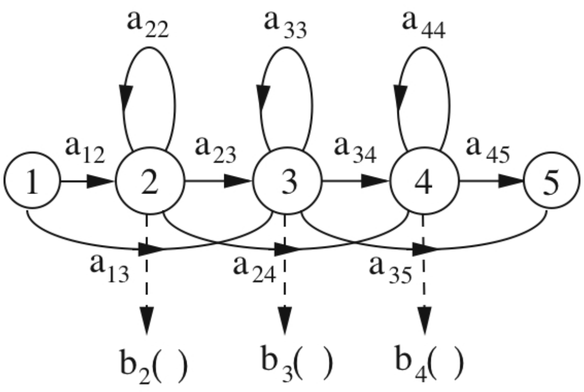


Figure 3.1: a typical HMM structure, consisting of 5 states, where the 1st and the 5th are not emitting states. a_{ij} are the transition probabilities from one state to the other, and b_k are the emission probabilities for a state k. This is a left to right HMM, that means an HMM that evolves only in one direction (no loops), with two skips (from state 2 to 4 and 3 to 5 respectively)

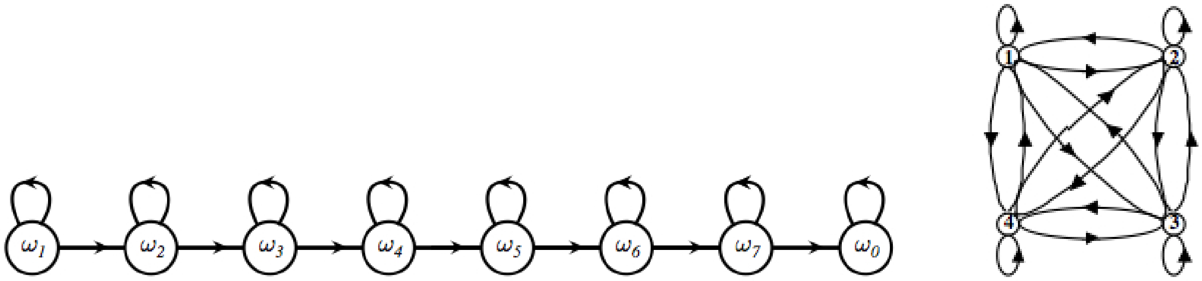


Figure 3.2: HMM topology: interconnection between states can result to different topologies. Here, a left-to-right HMM with no skips and loops and a more complex cyclic (ergodic) HMM

Particular types of HMMs that are of interest in gesture recognition are:

- *Continuous*: is the type of HMM where the symbols emitted do not belong to a discrete vocabulary of symbols. On the contrary, they can take stochastic values, i.e from a distribution function with mean M and variance σ.

- *Mutlidimensional* is the HMM that instead of one value, they can emit a vector of values instead of one when passing from emitting states.

As far as gesture recognition is concerned, a gesture is described by a set of N distinct hidden states and r dimensional M observable symbols. An HMM is characterized by a transition matrix A and r output distribution matrices $B_i, i = 1, ..., r$. Meaningful gestures may be very complex, containing simultaneous motions of a number of points. However, these complex gestures should be easily specifiable. In general, gestures can be specified either by example or by description. In the former, each application has a training session in which examples of different gestures are collected for training the models. The trained models are the representations of all gestures that the system must recognize. In the latter method of specification, a description of each gesture is written in a gesture description language, which is a formal language in which the syntax of each gesture is specified. Obviously, the example method has more flexibility than the description method. One potential drawback of specification by example is the difficulty in specifying the allowable variation between gestures of a given class. This problem would be avoided if the model parameters were determined by the most likely performance criterion. Because gesture is an expressive motion, it is natural to describe such a motion through a sequential model. Based on these considerations, HMM is appropriate for gesture recognition. A multi-dimensional HMM is able to deal with multi-path gestures which are general cases of gesture recognition. Moreover, a multi-dimensional HMM provides the possibility of using multiple features to increase the recognition rate.

The key idea of HMM-based gesture recognition is to use multi-dimensional HMM representing the defined gestures. The parameters of the model are determined by the training data. The trained models represent the most likely human performance and are used to evaluate new incoming gestures.

According to Yang,Xu[58], the HMM-based gesture recognition approach can be described as follows:

1. Define meaningful gestures - To communicate with gestures, meaningful gestures must first be specified. For example, a certain vocabulary must be specified for a sign language, and certain editor symbols must be given in advance if the gestures are to be used for editing text files.

2. Describe each gesture in term of an HMM - A multi-dimensional HMM is employed to model each gesture. Note that only the structures of A and B are determined in this step and the values of elements in A and B will be estimated in the training process.

3. Collect training data - In the HMM-based approach, gestures are specified through the training data. It is essential that the training data be represented in a concise and invariant form. Raw input data are preprocessed before they are used to train the HMMs. Because of the independence assumption, each dimensional signal can be preprocessed separately.

4. Train the HMMs through training data - Training is one of the most important procedures in a HMM-based approach. The model parameters λ are adjusted in such a way that they can maximize the likelihood $P(O|\lambda)$ for the given training data. That means the probability that the HMM gives an observation sequence O, given parameters λ. No analytic solution to the problem has been found so far. However, the Baum-Welch algorithm can be used to iteratively reestimate model parameters to achieve the local maximum.

5. Evaluate gestures with the trained model - The trained model can be used to classify the incoming gestures. The Forward-Backward algorithm or the Viterbi algorithm can be used to

classify isolated gestures and continuous gesture respectively, estimating the likelihood $P(O|\lambda)$ with reduced cost.

Chapter 4

The model

In this chapter we introduce 'Pogany', a tangible affective interface. Afterwards, an overview of the system for music interaction based on Pogany is presented. At the end of the chapter, we give a description for the front-end of the system.

4.1 Pogany: An affective facial interface

'Pogany' is a head-shaped tangible interface for the generation of facial expressions through intuitive contacts or proximity gestures [54]. It was conceptualized and constructed in LIMSI laboratory the year 2005 and is still in development. The purpose of the constructor is to offer a new medium of communication that can involve the user in an affective loop[56], [57]. The interface takes advantage of the existing non-expensive, integrated camera-capture technology, passing a video stream to a computer for processing. However, due to the design of the interface, only parts of the whole video image are worth to be processed. Hence the total amount of row video data to process is reduced from the beginning.

4.2 Description of the interface

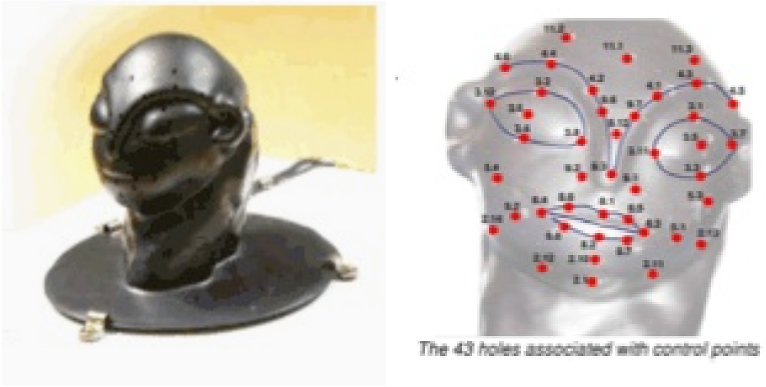


Figure 4.1: physical interface and list of interactive keypoints-holes

The input to the interface consists of intentional or/and natural affective gestures. We take advantage of the anthropomorphic shape of the input, a stylized human head, to establish easily learnable correspondences between users' contacts by hand and expressed emotions. As suggested by the designers in [54], the physical and input part of the interface is based on the following constraints:

- *it should be able to capture intuitive gestures through hands and fingers as if the user were approaching someones face,*
- *direct contacts as well as gestures in the vicinity of the head should also be captured in order to allow for a wide range of subtle interactions,*
- *the most expressive facial parts of the interface should be easily identified without the visual modality in order to allow for blinded interaction: eyes, eyebrows, mouth, and chin should have clearly marked shapes,*
- *as suggested by the design study of SenToy [59], the shape of the physical interface should not have strongly marked traits that would make it look like a familiar face, or that would suggest predefined expressions. (The evaluation has reports that the user qualify the face as expressionless, calm, placid...).*

Apart from the pure character of 'Pogany' as an interface for affective communication, what was investigated during this work was the appropriateness of such an interface for musical creation and interaction purposes. For an interface such as Pogany, such a task sounds challenging a priori basically for two reasons:

- The familiarity of a user with the human face, either by view or touch, can help the user associate instrumental gestures for the manipulation of the interface with common hand gestures he has experienced a lot of times before for non musical purposes. Thus, such a music interface can facilitate apprenticeship.
- Particular gesture patterns may correspond to high-level expressive or emotional information. For instance, if we regard a real human face as the interface itself, and we somehow detected the facial expressions produced by the alteration of the face parts (nose, lips, etc), we can then directly have a link between these expressions and corresponding emotions ([60]). For Pogany, apart from an association between facial expressions and emotions that the user can realize, additional emotional information can occur by the type and the particular area of the contact that the user can have with the interface. For instance, when someone touches a face on the cheek, depending on the force used and the speed and suddenness of the gesture, this action could be each time attributed to contradictory emotional intentions: from expression of calmness and tenderness to inelegance and brutality, with an extreme variability such as the one that exists between a caress and a punch! Such emotions would be a very interesting input in the design of a virtual music instrument. What remains a non-difficult task is the validation, classification and detection of such kind of emotions with the proper interface. Therefore Pogany, as a member of the affective interfaces family, has a major advantage against other interfaces to be used as a creative medium in an art concept, in our context, to be tested for music interaction purposes.

In the following chapters, and just after an overview of the global architecture of the system employed for Pogany to serve music application purposes, we will have a thorough look over the properties and the implementations concerning each part of the system: from the front-end, the feature detector and gesture recognizer, until the last-end, a music synthesizer.

4.3 Overview of Pogany as an expressive music interface

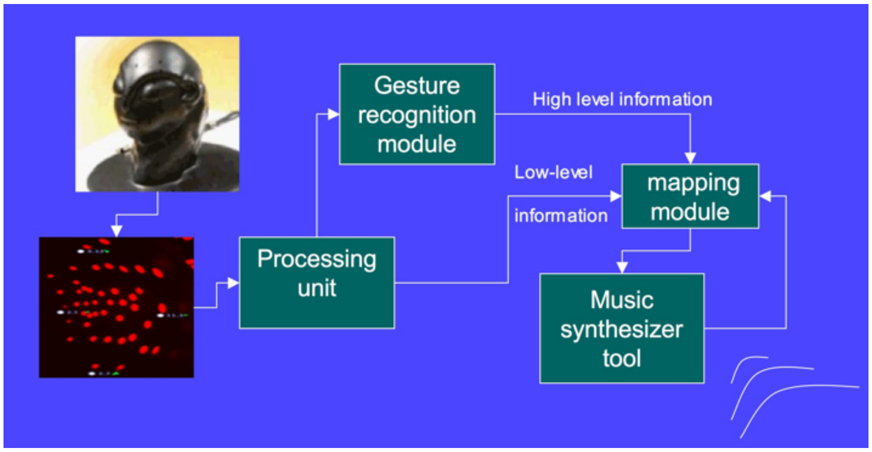


Figure 4.2: The overview of Pogany music interaction system's design

An overview of the system that controls Pogany's usage as a music synthesis interface can be viewed in figure 4.2. The scope of the work is to provide the user with a system the architecture of which suggests hybrid low and high-level techniques for gesture recognition and identification in order to capture expressivity. In the front part of this system, we first drag the important information from the row video data captured by the camera laying inside the interface. Therefore we isolate the important image pixel blocks in each frame. Then, in the middle part of the system, we process this information in a first-level approach, in order to extract the important features to use either directly for mapping or with a second higher layer of processing (Gesture Recognition Module). At the last part of the system and after a meta-processing procedure, we map these continuous or discrete feature values to the music synthesizer that produces the sound feedback of gestural action. In next sections we will thoroughly investigate the design issues arising for each part of the system and the strategies that have been followed.

4.4 Front part

The above constraints gave to the interface the form of figure 4.1, according to the interactive keypoints noted in the right part of the picture. These keypoints, with the form of small holes on the surface of the interface correspond exactly to a part of MPEG-4 compliant parameters (FAP's and FDP's). In a lit environment, passing over or covering the holes with the hands variates the luminosity level captured by a camera placed in the facial interface. From each frame of the raw video image captured by the camera we drag only the pixel blocks that correspond to the important keypoints, i.e the names where the holes are positioned on the surface of 'Pogany'. At the end, the mean value of luminocity for every keypoint pixel block is collected to a feature vector. The output of the front-end of the system consists of instantiations of a 43 element vector with a rate of 30 fr/sec, thus providing the HMM core with a low dimensional vector instead of raw data of image format.

The front-end module of the system, is based on the use of a camera and a proper video-capture software. At system already built in LIMSI, real-time correspondences between gesture and keypoints were established. An image segmentation tool keeps only the blocks of major importance and finds the normalized mean luminocity value of the pixels that belong to each block. In this way we keep just one value of light intrusion for each of the pixel blocks that correspond to the holes. The metric that we will use for processing is called alpha value, defined as:

$$alpha\ value = \frac{current\ luminocity}{luminocity\ at\ calibration\ time} \tag{4.1}$$

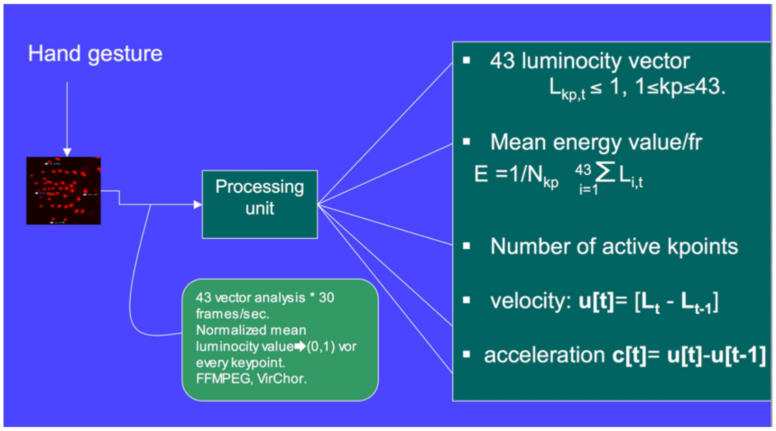
Alpha value is bounded between 0 and 1, with 0 corresponding to maximum light intrusion (that means no covering of the hole, thus zero activity) and 1 to minimum light intrusion (the hole is fully covered, maximum activation of the KeyPoint). Then, an image calibration tool to adjust the position of the 43 image segments according to manual orientation of only 4 dominant segments (KeyPoints) and the rendering of the feature vector with a 30 fr/sec framerate was provided through VirChor software environment[61]. Further information about the particular techniques used can be found in [54].

Chapter 5

Middle part

The middle part of the system includes the processing-feature extraction unit and the gesture recognition module. In the first, we extract useful information to drive the music synthesis procedure directly: This low-level information, apart from being used in the framework of direct mapping strategies described in 6.3.1, it also provides the input for the gesture recognition module. The last is responsible for providing a layer that recognize higher-level information arising from the gestures of the user: this information concerns expressivity-related commands that tend to modify the music synthesis procedure in the form of modulation or interrupts.

5.1 Gesture Analysis-feature extraction



Hand gesture

Processing unit

43 vector analysis * 30 frames/sec.
Normalized mean luminocity value➡(0,1) vor every keypoint.
FFMPEG, VirChor.

- 43 luminocity vector $L_{kp,t} \leq 1$, $1 \leq kp \leq 43$.
- Mean energy value/fr $E = 1/N_{kp} \sum_{i=1}^{43} L_{i,t}$
- Number of active kpoints
- velocity: **u[t]= [L_t - L_{t-1}]**
- acceleration **c[t]= u[t]-u[t-1]**

Figure 5.1: The feature extraction unit

This module is responsible for the first-level feature analysis and extraction 5.1. The input to this module is a 43-size vector with of 30 instances/second, coming from the image segmentation procedure in the front-end part. Every element of this vector has a value between 0-1, which correspond to the mean luminocity value detected for the pixels that belong to each particular pixel-block of concern.

The primal feature that were selected for such a procedure were:

Energy Paticular meaning for the mapping procedure in next stage has the definition of a measure for the energy of the signal. This 43-dimensional signal, with values normalized between 0 and 1 (0 for full light penetration and 1 for zero light detection for each of the holes of the interface),thus, a energy measure for the signal denotes activation in front of the interface. We call this multidimensional signal $\mathbf{X}_t$. In case we define energy E_t as:

$$\mathbf{E_t} = \mathbf{X}_t^2, \tag{5.1}$$

where $\mathbf{E_t}$ is the temporal energy vector for the frame t=0, 1,..n. The energy value for each Keypoint i at frame t can then be found by: $E_{i,t} = X_{i,t}^2$, where $i = 1..N_{kp}$ the N_{kp} different KeyPoint's value that constitute the vector.

Hence, the normalized mean short time energy of the signal at frame t is:

$$\overline{E_t} = \frac{1}{N_{kp}} \sum_{j=1}^{N_{kp}} {X_{j,t}}^2, \tag{5.2}$$

As the signal does not take negative values, it is not wrong instead of energy to consider $\overline{M_t} = \sum_{j=1}^{N_{kp}} X_{j,t}$, where M_t the *Mean Magnitude Value per frame*, as a metric for the activation of the KeyPoints of the interface. It is straightforward that this metric implicitly represents the general amount of activation in the vicinity of the interface: therefore, it will be used in the segmentation procedure and in particular the detection of either some kind of activity (gestural or postural) or, for values near zero, 'gesture silence'. This is further described in 5.3.2.

Velocity The velocity of the multidimensional signal is defined as:

$$\mathbf{V}_t = \frac{\mathbf{X}_t - \mathbf{X}_{t-\delta t}}{\delta t}, \; t = 1, 2, ..n. \tag{5.3}$$

We assume that $\mathbf{V}_0 = 0$;

If $\delta t = 1$, the velocity value for each keyPoint is:

$$V_{i,t} = X_{i,t} - X_{i,t-1} \tag{5.4}$$

The mean velocity value per frame t is then:

$$\overline{V_t} = \frac{1}{N_{kp}} \sum_{j=1}^{N_{kp}} X_{j,t} - X_{j,t-1}, \tag{5.5}$$

where $i = 1..N_{kp}$, $t \geq 1$.

A useful measure to be used for gesture segmentation is the *Mean Activation Rate*:

$$M.A.R_t = \frac{1}{N_{kp}} \sum_{j=1}^{N_{kp}} |X_{j,t} - X_{j,t-1}|, \tag{5.6}$$

where $i = 1..N_{kp}$, $t \geq 1$.

In figure 5.2 we show several graphs for the previous features Mean Magnitude per Frame and Mean Activation Rate per Frame. These graphs correspond to the gestures 'eyclose' and 'smile' (gestures are defined in the next section). For these two particular gestures the hand of the user

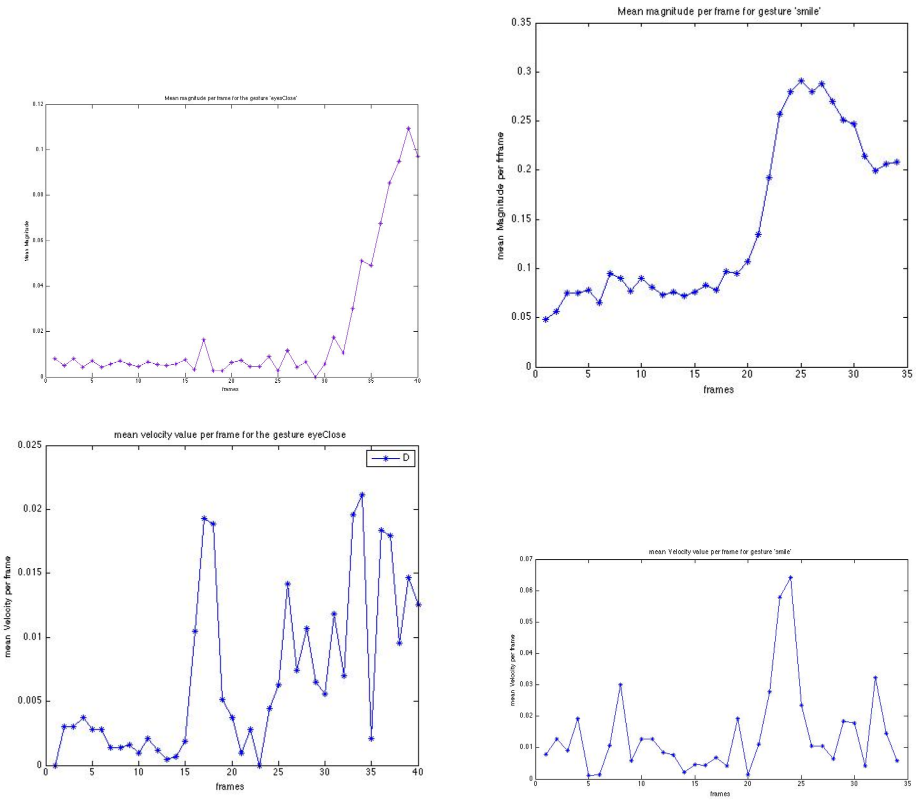


Figure 5.2: Mean Magnitude per frame and Mean Activation Rate per frame for two different gestural interactions with the interface: a)on the left with the gesture 'eyeclose' b)on the right with the gesture 'smile' (for the type of gestures see figure 5.3)

.

remains in touch with the interface. Even though these values do not provide themselves enough evidence of the gesture made, so as to use for gesture identification, however they give important clues for the amount of gesture activity as well as the smoothness -or suddenness- of a gesture (see 5.3.2). Furthermore, through the Mean Activation Rate it is possible to detect the intervals of motionlessness between two sequential gestures. This has be proved to be useful for Pogany recognition system, more specifically during recognition of continuous gesture.

5.2 Gesture Recognition module

The gesture recognition module is responsible for the identification of the gesture that the users implies to the interface. In a first approach, it was worth defining a) the recognition system that should be used in such a scope and b) the estimation of the parameters of such a pattern recognition system in order to optimize its performance.

5.2.1 HMM machine

One of the advantages of the HMM-based approach is that a variety of knowledge sources can be combined into a simple HMM. A recognition model can be simply created by putting all gesture models in parallel, and adding an initial state and a final state[58]. A similar approach is to construct a number of HMM equal to the number of isolated gestures that we want to recognize, and the $P_i(O|\lambda)$, where $i$the identifier of the model corresponding to an isolated gesture. Beyond this selection, when constructing an HMM network, there are several questions arising from the nature of the procedure represented by the HMM that need to be answered in order to achieve the best performance possible. This consist of:

- the number of physical states N to choose for the model. This remains a value to be selected -or at least checked- manually.
- the appropriate HMM topology should be also manually chosen. Should it be a triangle, or a left to right, with skips, or loops?
- the mean and variance initialization, if it is for a continuous HMM. There is no robust algorithm to optimize initialization. Though the Expectation- Maximization algorithm converges, it strongly depends on the initial values whether it will converge to the global or the local maximum.
- the size of the observation vector and the behavior of the multidimensional HMM. One of the most important tasks is to make the selection of the most salient and non-redudant features that can get extracted from the image.
- the number of the training samples to train each HMM.
- the construction of a proper vocabulary to serve the needs of continuous recognition.
- the strategies to follow for continuous gesture recognition, that means to define a proper grammar in order to specify to the system in which way gestures follow one the other. An other issue is the segmentation of the continuous stream of gestures. Constraints in gesture series resulted to constrain programming techniques in order to enhance the recognition procedure.

Our first part of research included 2 experiments, both for isolated gesture recognition. This decided the configuration for the real-time recognition toolkit.

5.2.2 Experiments for gesture recognition and strategies for HMM configuration

First experiment In the first experiment, we trained a 'left to right HMM' with data form 6 gesture categories: *down-up*, *up-down*, *eyebrows-lifting*, *eye-closing*, *getting sad*, *smile* (figure 5.3) . The first two gesture categories correspond to the tangible contact throughout the face in two with different direction each time. The names of the gesture types and the categorization criteria had no particular cognitive meaning: it was rather inspired from the emoticons' experiment in [54], describing the simplified passage from a neutral to a biased face condition. In other words, gestures committed concerned the simplified animation of certain face characteristics, in order to obtain a biased emoticon-like facial expression [60]. Thus, the $eyebrows - lifting$ is the tangible gesture tending to lift up the eybrows, the $eye - closing$ to close the eyes of the interface, and similarly *getting sad* and *smile* express gestures that tend to modify Pogany's mouth-line analogously. Finally *static* corresponds to the motionless continuous touch on the mouth area of the interface.

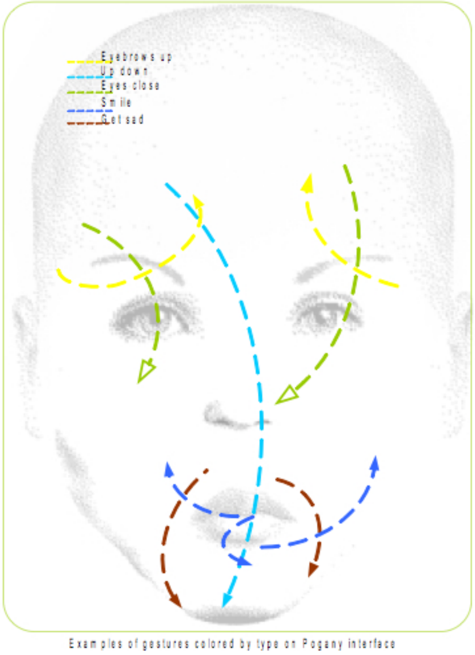


Figure 5.3: examples of gestures used as data for training and testing during the experiments

On the first experiment we made use of 142 total samples, taken under full, but slightly non-homogenous lighting conditions. Cross validation method was used, so as 2/3 of the total samples were used for system training, and the other 1/3 for evaluating the system. The HMM to train was a 5-state left-to-right continuous multi-dimensionnal HMM, with observations vector of size 43. The interface had an orientation of zero angle with the user.

Second experiment At the second experiment, the number of samples used was 90, but the gestures to recognize have become four: *eyebrows-lifting*, *eye-closing*, *getting sad*, *smile*. We used cross-validation method for training and evaluating system performance. This time the experiment included a test-phase, under different lighting conditions. To specify, while all 90 samples were collected under homogenous artificial light conditions, test data(totally 70 gestures) were taken under full physical light conditions, slightly brighter on the right of 'Pogany' than on the left. The

HMM prototype used was similar to this of the first experiment, except for the number of states that this time was set to 4.

5.2.3 results

Below we present the results of two experiments, in the form of confusion matrices: along the vertical axis we have the real classes where each gesture belong and along the horizontal axis the classification output of the recognizer.

Table 5.1: 1st experiment: Confusion matrix

	down2UP	up2D.	eyebrUP	eyeCL	getSAD	SMILE	stat.	succ.
down2UP	1	1	0	0	0	0	0	50 %
up2DOWN	0	3	0	0	0	0	0	100%
eyebrowsUP	0	0	19	1	0	0	0	95 %
eyeCLOSE	0	0	0	11	0	0	0	100%
g.SAD	0	0	0	0	9	0	0	100%
SMILE	0	0	0	0	0	12	0	100%
static	0	0	0	0	0	0	5	100%
TOTAL								96.77%

Table 5.2: 2nd experiment: Confusion matrix

	eyeCLOSE	eyebrowsUP	getSAD	SMILE	success
eyeCLOSE	9	0	0	0	100 %
eyebrowsUP	0	10	0	0	100 %
getSAD	0	0	9	0	100 %
SMILE	0	0	0	7	100 %
TOTAL					100 %

Table 5.3: 2nd training set , testing set from different conditions

	eyeCLOSE	eyebrowsUP	getSAD	SMILE	success
eyeCLOSE	11	2	1	0	78.5 %
eyebrowsUP	0	15	0	0	100 %
getSAD	0	0	26	2	92.85%
SMILE	0	0	5	7	58.33%
TOTAL					84.39 %

At both experiments the results we received were more than encouraging. System performed 96.77% and 100% experiments respectively, proving that both the interface and HMM correspond nearly perfectly to the validation procedure. Taking into consideration random selection of samples through cross-validation method, we dare to say that the isolated gesture recognition system exhibits a high-level performance for steady light conditions (validation results). The 3rd matrix reveals several weakness of the system to adapt in different lighting environments, its performance though remains tolerable 84.37%.

Although fail in recognition can be attributed to the drawback of HMM to converge to local minima[58], probabilities estimated through the testing process in second experiment prove a small win for the wrong HMM over the correct ones (mean error: 4,7%), creating the expectation to improve results under careful modelisation and HMM initialization.

5.2.4 Conclusions for isolated gesture recognition

Results presented in the confusion matrices show that the problems the system encounters correspond to gestures sharing the similar area on the interface. Particularly for the recognition of 'smile' in the test phase of the second experiment, all wrong classified gestures were confused with 'sad'. An other type of confusion happens when two gestures have the same directivity (smile at matrix 5.3)

In order to get advantage from the system's good performance isolated gesture recognition , we were motivated to approach recognition of continuous gestures by segmenting the continuous stream to intervals of motionlessness (hands of the user remain unmoved). Enhancing certain aspects of the 'Pogany' interface (quality of camera and speed of connection), could further improve general performance in both gesture recognition and precise, effective interactivity.

Results shown that HMM, through certain manipulations and enhancements, could provide 'Pogany' a gesture recognition module that decodes expressive information. On the next step of the research we concentrate on the construction and evaluation of a model for real-time recognition of complex gestures classified according to face emotions. High-level indirect information will then be mapped to a music synthesis tool in order to control the evolution of implicit music events and data-driven music synthesis.

5.3 Real-time recognition for Pogany

5.3.1 Overview

The experiments with HMM recognition for the Pogany interface showed that an HMM configuration could lead, in a first place, rather easily to the recognition of a reasonable amount of gestures. However, the requirements of the interface in our case are set from the purposes of an interface for music interaction. Thus, the desirable would be to add to the system a module that recognizes real-time gesture patterns on the vicinity of the interface. Inspired from our experiments for off-line isolated gestures based on HMMs, we developed a real-time module for continuous gesture. On the parallel, we were interested in keeping a high degree of expandability for the system, that means to let open future enhancements with multiple gestures, complex gestures and a large-scale gesture vocabulary. In the following subsections, we will describe more profoundly the strategies used for the gesture segmentation, supervised training, and real-time recognition of postures and gesture intention identification.

An overview for the gesture recognition tool implemented for real time can be viewed in figure 5.4. This real-time module is destined to recognize gestures of continuous contact with the interface. On the top we can see the activity detector module: it is responsible for the detection (or not) of gesture activity in the front of the interface. In case of gesture of posture, the HMM recognizer is activated, bound each time with a proper vocabulary arising from training over a specific gesture or posture corpus.

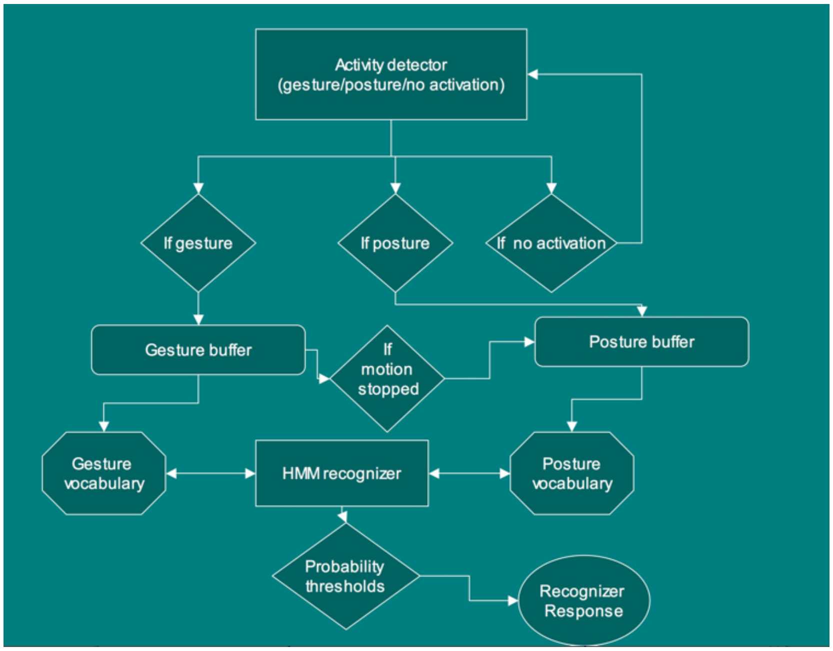


Figure 5.4: Real-time Recognition flow

5.3.2 gesture segmentation

Segmentation of gesture is implemented on the activity detector module. This module is responsible of segmenting the row data on gestures, postures, and silence parts (referenced as 'no activity' in figure 5.4). For this scope, features such as *Mean Magnitude Value per frame(MMV)* and *Mean activation Rate per frame(MAR)* are employed.

discriminating gesture or posture from silence More specifically *Mean Magnitude Value*, as mentioned in section 5.1, MMV is a metric for gesture activity on the vicinity of the interface: it can define whether there is activation in front of the interface or not (as shown also from figure 5.2). It was found that a threshold $0 \leq thres \leq 0.3$, regulates the activation decision of the activity detector module(5.4): If $MMV > thres$, then there is gesture or posture, if not, there is silence (5.5).

To enhance stability in noise situations, an additional tool counts the number of KeyPoints which are activated: with the term 'activated' we mean the KeyPoints that have an *alpha value* over a certain threshold. If the Mean Magnitude Value shows that we have an activation (gestural or postural) and on the same time the number of activated keypoints is less than a certain number, this increases the possibilities of a uniformly distributed noise all over the critical surface area of the interface. In effect, in this situation there is no gesture, this can be detected by the second check through the number of the activated KeyPoints. Thus, the number of activated KeyPoints is correlated with the Mean Magnitude Value per frame in order to optimize system's activity detection performance.

decision between gesture or posture The segmentation system also demands the discrimination between gestures and postures. In this case, Mean Magnitude Value per Frame cannot distinguish the difference between the two, as there is activation in both cases.

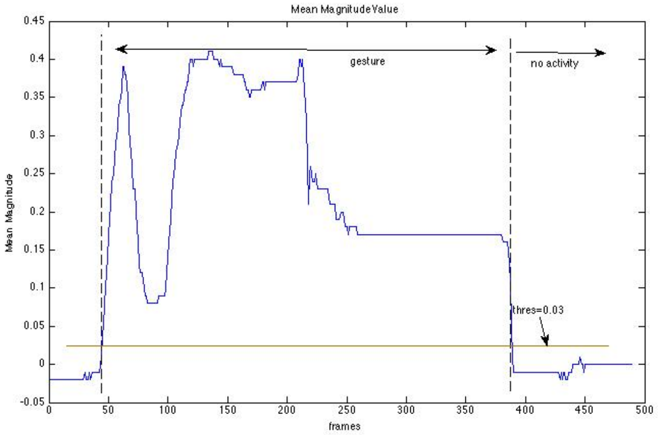


Figure 5.5: Mean Magnitude Value during activation and no activity intervals.

The selection between these two different states (gestures and postures) can be efficiently expressed by the *Mean Activation Rate(MAR)* metric. As described in section 5.1, the MAR value is just a variated expression for gesture velocity. Thus, a MAR close to zero expresses static activity, where a great value of MAR suggest a rapid gesture. Thus, being over a certain threshold lets suppose $MAR > thres'$, we certify gesture activity. Below this threshold, the Mean Activation Value signifies a posture. In figure 5.6 we wan observe how Mean Activation Rate value is decreasing, signifying the end of the gesture and possibly the start of a posture.

In order to identify gestures or postures in between raw data, a the gesture activation tool is first employed. In case there is evidence for activation, the Mean Activation Rate decides if it cases for a gestural or postural activity.

5.3.3 HMM configuration

Inspired from the experiments for isolated gesture recognition, the number N of states for the HMM was set to 4, plus the two non-emitting states at start and at the end. The topology remained as left-to-right with no skips. The size of the observation vector was 43, and the input to this vector were the alpha values of the corresponding keypoints. The gestures that the system was trained to recognize were six (two more than the gestures described in the second experiment of the previous section), and the number of postures was was also 6 (revealing steady positions at the area of the eyes, eyebrows, nose, cheeks, mouth and chin). The system used switching grammars in order to choose either from a gesture or a posture vocabulary.

5.3.4 Training

The training of the system was based on data from one user. The gesture samples used for training were15 for each gesture or posture. The training of the system was based on the Baum-Welsh algorithm. In our case, which is the continuous HMM, rules must be set for the estimate the means and variances of a HMM: in order to intialize in the first place and to re-estimate at the end of each repetition. We assume that our observation follows the single component Gaussian distribution:

$$b_j(\mathbf{o}_t) = \frac{1}{\sqrt{(2\pi)^n|\mathbf{\Sigma_j}|}} e^{-\frac{1}{2}(\mathbf{o}_t-\mu_j)'\mathbf{\Sigma}_j^{-1}(\mathbf{o}_t-\mu_j)} \tag{5.7}$$

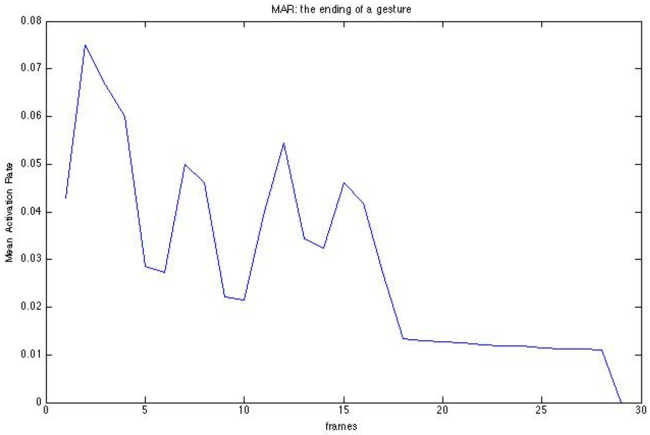


Figure 5.6: Mean Activation Rate at the finish of a gesture. As the Rate falls under a certain threshold, it triggers the end of buffering for the gesture, the recognition of the gesture and the start of posture buffering.

Since the full likelihood of each observation sequence is based on the summation of all possible state sequences, each observation vector $\mathbf{o}_t$ contributes to the computation of the maximum likelihood parameter values for each state j. In other words, instead of assigning each observation vector to a specific state as in the above approximation, each observation is assigned to every state in proportion to the probability of the model being in that state when the vector was observed. Thus, if $L_j(t)$ denotes the probability of being in state j at time t for the weighted averages we have:

$$\hat{\mu}_j = \frac{\sum_{t=1}^{T} L_j(t)\mathbf{o}_t}{\sum_{t=1}^{T} L_j(t)} \tag{5.8}$$

and

$$\hat{\mathbf{\Sigma}}_j = \frac{\sum_{t=1}^{T} L_j(t)(\mathbf{o}_t - \mu_j)(\mathbf{o}_t - \mu_j)'}{\sum_{t=1}^{T} L_j(t)} \tag{5.9}$$

where the summations in the denominators are included to give the required normalisation.

Equations 5.8 and 5.9 are the Baum-Welch re-estimation formulae for the means and covariances of a HMM. The probability of state occupation $L_j(t)$ is calculated. using the *Forward-Backward* algorithm. Let the forward probability $\alpha_j(t)$ for some model M with N states be defined as

$$\alpha_j(t) = P(\mathbf{o}_1, \ldots, \mathbf{o}_t, x(t) = j|M). \tag{5.10}$$

That is, $\alpha_j(t)$ is the joint probability of observing the first t vectors and being in state j at time t. This forward probability is calculated by the following recursion

$$\alpha_j(t) = \left[\sum_{i=2}^{N-1} \alpha_i(t-1)a_{ij}\right] b_j(\mathbf{o}_t). \tag{5.11}$$

This recursion depends on the fact that the probability of being in state j at time t and seeing observation $\mathbf{o}_t$ can be deduced by summing the forward probabilities for all possible predecessor states i weighted by the transition probability a_{ij}. The slightly odd limits are caused by the fact that states 1 and N are non-emitting. The initial conditions for the above recursion are

$$\alpha_1(1) = 1 \tag{5.12}$$

$$\alpha_j(1) = a_{1j} b_j(\mathbf{o}_1) \tag{5.13}$$

for $1 < j < N$ and the final condition is given by

$$\alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T) a_{iN}. \tag{5.14}$$

Notice here that from the definition of $\alpha_j(t)$,

$$P(\mathbf{O}|\lambda) = \alpha_N(T). \tag{5.15}$$

The backward probability $\beta_j(t)$ is defined as

$$\beta_j(t) = P(\mathbf{o}_{t+1}, \ldots, \mathbf{o}_T | x(t) = j, M). \tag{5.16}$$

For the backward probability we have similarely:

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_j(t+1) \tag{5.17}$$

with initial condition given by

$$\beta_i(T) = a_{iN} \tag{5.18}$$

for $1 < i < N$ and final condition given by

$$\beta_1(1) = \sum_{j=2}^{N-1} a_{1j} b_j(\mathbf{o}_1) \beta_j(1). \tag{5.19}$$

From the definitions,

$$\alpha_j(t)\beta_j(t) = P(\mathbf{O}, x(t) = j | M). \tag{5.20}$$

Hence,

$$\begin{aligned} L_j(t) &= P(x(t) = j | \mathbf{O}, M) \\ &= \frac{P(\mathbf{O}, x(t) = j | M)}{P(\mathbf{O}|\lambda)} \\ &= \frac{1}{P} \alpha_j(t) \beta_j(t) \end{aligned} \tag{5.21}$$

where $P = P(\mathbf{O}|\lambda)$.

The steps in this algorithm may be summarised as follows

1. For every parameter vector/matrix requiring re-estimation, allocate storage for the numerator and denominator summations of the form illustrated by equations 5.8 and 5.9. These storage locations are referred to as *accumulators*. Note that normally the summations in the denominators of the re-estimation formulae are identical across the parameter sets of a given state and therefore only a single common storage location for the denominators is required and it need only be calculated once.

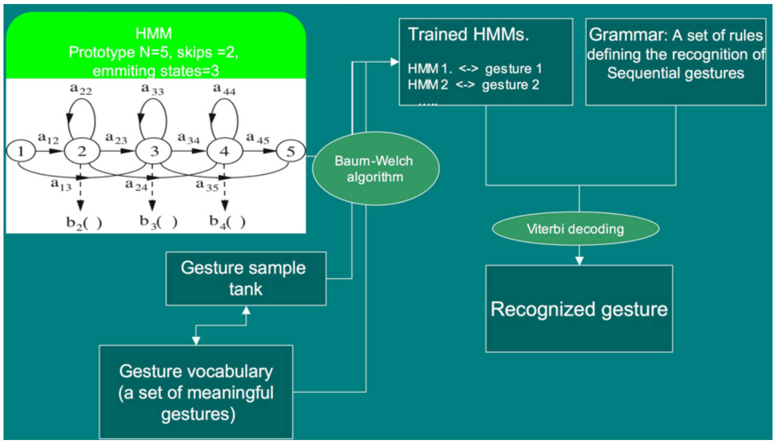


Figure 5.7: HMM recognition module

2. Calculate the forward and backward probabilities for all states j and times t.

3. For each state j and time t, use the probability $L_j(t)$ and the current observation vector $\mathbf{o}_t$ to update the accumulators for that state.

4. Use the final accumulator values to calculate new parameter values.

5. If the value of $P = P(\mathbf{O}|\lambda)$ for this iteration is not higher than the value at the previous iteration then stop, otherwise repeat the above steps using the new re-estimated parameter values.

All of the above assumes that the parameters for a HMM are re-estimated from a single observation sequence, that is a single example of a gesture. In practice, many examples are needed to get good parameter estimates. However, the use of multiple observation sequences adds no additional complexity to the algorithm. Steps 2 and 3 above are simply repeated for each distinct training sequence.

One final point that should be mentioned is that the computation of the forward and backward probabilities involves taking the product of a large number of probabilities. In practice, this means that the actual numbers involved become very small. Hence, to avoid numerical problems, the forward-backward computation is computed in using log arithmetic.

5.3.5 Real-time Recognition: Viterbi decoding

The architecture of recognition can be seen in figure 5.7.

The previous paragraph has described the basic ideas underlying HMM parameter re-estimation using the Baum-Welch algorithm. In passing, it was noted that the efficient recursive algorithm for computing the forward probability also yielded as a by-product the total likelihood $P(\mathbf{O}|\lambda)$. Thus, this algorithm could also be used to find the model which yields the maximum value of $P(\mathbf{O}|\lambda_i)$, and hence, it could be used for recognition.

In practice, however, it is preferable to base recognition on the maximum likelihood state sequence since this generalises easily to the continuous gesture case whereas the use of the total probability does not. This likelihood is computed using essentially the same algorithm as the forward probability calculation except that the summation is replaced by a maximum operation. For a

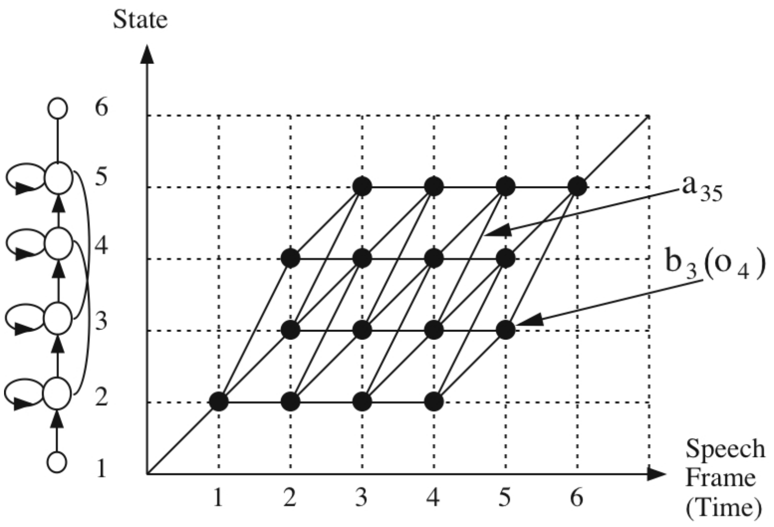


Figure 5.8: A trellis structure corresponding to the iteration of Viterbi Algorithm in a speech recognition case

given model λ, let $\phi_j(t)$ represent the maximum likelihood of observing gesture vectors $\mathbf{o}_1$ to $\mathbf{o}_t$ and being in state j at time t. This partial likelihood can be computed efficiently using the following recursion (cf. equation 5.11)

$$\phi_j(t) = \max_i \{\phi_i(t-1)a_{ij}\} \, b_j(\mathbf{o}_t). \tag{5.22}$$

where

$$\phi_1(1) = 1 \tag{5.23}$$

$$\phi_j(1) = a_{1j} b_j(\mathbf{o}_1) \tag{5.24}$$

for $1 < j < N$. The maximum likelihood $\hat{P}(O|\lambda)$ is then given by

$$\phi_N(T) = \max_i \{\phi_i(T)a_{iN}\} \tag{5.25}$$

As for the re-estimation case, the direct computation of likelihoods leads to underflow, hence, log likelihoods are used instead. The recursion of equation 5.22 then becomes

$$\psi_j(t) = \max_i \{\psi_i(t-1) + log(a_{ij})\} + log(b_j(\mathbf{o}_t)). \tag{5.26}$$

This recursion forms the basis of the so-called Viterbi algorithm. As shown in figure 5.8, this algorithm can be visualised as finding the best path through a matrix where the vertical dimension represents the states of the HMM and the horizontal dimension represents the frames of gesture data (i.e. time). Each large dot in the picture represents the log probability of observing that frame at that time and each arc between dots corresponds to a log transition probability. The log probability of any path is computed simply by summing the log transition probabilities and the log output probabilities along that path. The paths are grown from left-to-right column-by-column. At time t, each partial path $\psi_i(t-1)$ is known for all states i, hence equation 5.26 can be used to compute $\psi_j(t)$ thereby extending the partial paths by one time frame.

5.3.6 Implementations for off-line training and real-time recognition

The recognition model is based on HTK [55] library and toolkit. HTK is a toolkit for building Hidden Markov Models (HMMs). HTK offers a variety of tools to implement HMMs with numerous configuration capabilities. However, HTK is primarily designed for building HMM-based speech processing tools, in particular speech recognizers. As much of the infrastructure support in HTK is dedicated to this task, the use of the HTK library for gesture is not straightforward. Furthermore, the non object-oriented form of the code HTK library does not facilitate the code modifying approaches. Besides that, HTK has tools for off-line recognition, which means that in order to use with real-time recognition HTK should be properly integrated.

The modules that were intergrated were:

- An algorithm to collect training data
- Algorithm to transform the text data into the HTK-compatible binary labeled format.
- An algorithm to train the HMM through the use of different data categories, HMM configurations, vocabularies and grammars.
- A gesture/silence/posture detector, in conjuction with an implementation for continuous recognition
- A real-time gesture recognition module
- The base of real-time recognition module for gesture intentions, through the use of the Token-Passing algorithm

Collecting Data the algorithm for the data collection used for training is implemented within the Virtual Choreographer environment. The user is first asked to name the gesture he intends to train the system with. Then, in an automatic procedure, he performs sequential repeats from the same gesture. Data is first written as sequential float values in a form of text. Each vector consists of 43 such values, and seperates from vectors at different frames through the end line character.

Trasforming the data The HTK uses a self-defined binary label-format. A C++ file outside Virchor code was written based on the gt2k script collection in order to make the transform from a normal text or binary format to the HTK file format. This file is responsible of applying the 4-byte label at the beginning of each sequence to recognize. Then, byte swapping is performed, in order to get in alliance with the HTK's byte decoding order

HMM training A script (training.sh) in shell language performs all the training procedure for the HMM. This procedure takes place off-time, and gets use of the HTK- toolkit. To be more specific, the tools used are:

- HParse This executable is responsible for parsing the grammar and create the word(gesture) level network.
- HCompV, HInit, HRest HCompV fills in our mean and variances on the HMM model provided. When this option is set, HCompV updates all the HMM component means with the sample mean computed from the training files. HInit performs an initial estimation of the HMM

parameters by calculating mean values arising from the training files. Finally the HRest performs embedded training.

In the case of simple gestures, this mechanism performs the simultaneous training for every instantiation of the HMM (corresponding to each meaningful gesture). In effect, the potential of tool for embedded training gets far more than this: in the case of a complex gesture, this can be represented by the concatenation of simple gestures: Embedded training then is responsible for creating the links between the different HMM according to a certain topology specified. This metaphor is inspired from the diphone and triphone models for speech recognition, and performs a training process without the need of inter-word segmentation in phonemes. Therefore, it is very important in case of a large scale vocabulary divided in smaller construction elements, such as the phonemes are for words, the word for phrases etc.

The utility of embedded training was one of the more important factors for selecting the HTK library and transcribing it for real-time gesture recognition. Through such tools, it is possible to build under a set of constraints an entire network of gestures

- HVite This tool performs off-line Viterbi decoding recognition. It uses label files in order to make transcriptions from the training data and alliances between the real gesture identity and the estimation of the HMM recognition system. With HStats tool results are obtained.

A gesture/silence/posture detector This tool, which use was explained in previous paragraph, is directly implemented in C++ and is integrated in the VirChor environment.

A real-time gesture recognition module The real-time recognition module was linked the HTK internal library files HNet.c and HRec.c. Its code is based on the HVite's executable code; however, as declined by the HTK developers, this tool is not dedicated to work for real-time. Therefore, in order to adapt it to real-time purposes, we had to perform an amount of modifications in the HNet.c and Hrec.c files of the library.

Identification of gesture intentions In the case where the gesture to recognize is more complex and needs some time to be completed, it would be useful for interactivity issues to have some clues of the gesture permitted at the time it is made. However, this conflicts with the Viterbi algorithm recursive nature. A variation of the Viterbi algorithm, the Token Passing Algorithm can solve this problem. This algorithm is implemented in the HTK library. After proper modifications in the FViteproc.cpp file in VirChor it permits to make estimations in between frames, and under certain circumstances-that means when the recognition probability is over a certain threshold- to adjudge an estimation for the identification of the gesture performed.

The token passing model makes the concept of a state alignment path explicit. Imagine each state j of a HMM at time t holds a single moveable token which contains, amongst other information, the partial log probability $\psi_j(t)$. This token then represents a partial match between the observation sequence $\mathbf{o}_1$ to $\mathbf{o}_t$ and the model subject to the constraint that the model is in state j at time t. The path extension algorithm represented by the recursion of equation 5.26 is then replaced by the equivalent *token passing algorithm* which is executed at each time frame t. The key steps in this algorithm are as follows

1. Pass a copy of every token in state i to all connecting states j, incrementing the log probability of the copy by $log[a_{ij}] + log[b_j(\mathbf{o}(t)]$.

2. Examine the tokens in every state and discard all but the token with the highest probability.

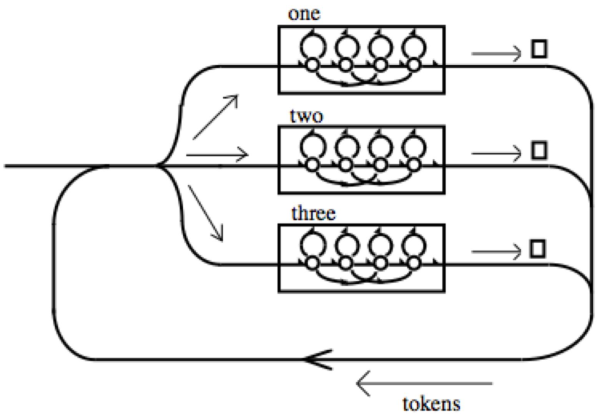


Figure 5.9: Here, three different HMMs merged on a single composite HMM. The tokens advance in parallel through the left-to-right HMMs. The difference is that we can have the winning token for the whole network after each algorithm iteration t.

Considering all HMMs representing different gestures as a composite HMM with the HMMs in parallel, it all tokens to advance simultaneously (5.9). In our case, discarding all but one token per state after each iteration of the algorithm, we achieve an estimation for the best tokens in all network. Using normalized forward possibilities, and taking into account multiple wins for tokens of the same HMM in time, we can achieve a suboptimal estimation of the gesture performed from the first moments of its occurance. As the real-time framework for gesture intention recognition is now set, it will allow further research on the subject.

Chapter 6

Mapping and music synthesis strategies

6.1 An overview on strategies for effective interaction

After a gesture is recognized, it can be used as a high-level emotional message from the user. The system, in order to create the conditions for an effective dialog, should then be able to produce a response containing information suitable with respect to the context and as much high-level as the users inputs. The strategy followed for a multimodal system is to analyze the user's gesture and create a semantic representation, then prepare a semantic response to the input according to a set of rules, and then map it to a sound synthesizer. Thus, if the semantic content of the analyzed gesture is classified somewhere in the *expressive semantic space*, the system can then reply with an element of the same cluster, in order to emphasize the emotional content of the gesture, or select an other region in the semantic space (i.e a neutral or contradictory emotion from the one received) to smoothen or resist to the user's input. In case this affective interaction concerns gesture and music, the system employs metaphors to stimulate music synthesis functions or control music parameters according to the expressive content of the received gesture. In [4], such strategies are discussed, as well as their combination with *expressive direct strategies*; that means strategies that are often associated with the lower levels of the conceptual framework. For 'Pogany', we employ a combination of both direct and indirect mapping strategies. Both will be discussed further in the following sections.

6.1.1 direct mapping strategies

The 43-feature vector used for the recognition module provides continuous information concerning activity in front of the interface. Apart from establishing one-to-one mapping connections among some feature elements and music parameters, other meaningful direct mapping techniques suggest a prior segmentation of the feature vector to smaller blocks with respect to the particular facial area they derive from (eyes, mouth, etc), followed by many-to-one mapping application. Useful additional features are obtained through estimation of the energy and velocity of the vector signal, which correspond to the horizontal axis, as well as the relative value of the energy among the facial areas. In a first approach, these values are metaphorically related with loudness and spectral modification respectively.

6.1.2 indirect mapping strategies

However, the HMM approach as a medium to capture semantic content, can also allow a higher-level mapping. Operating real-time HMM recognition on a layer above the direct mapping techniques can trigger more complex procedures of music parameter modification each time one or a series of gestures is recognized. Furthermore, with respect to the emotional content of gestures to the interface, it sounds inspiring to create correspondences between them and relevant or contradictory music procedures, that could establish an affective interaction. This can be achieved through concatenation of music structures, which are pre-classified according to perceptual criteria. However, the mapping of such information to music is a rather non-trivial task. First, because it is difficult to define a music emotional symbolic space that can be isomorphic to emotional gesture semantic space, due to the ambiguity of music emotional perception phenomenon. Second, in case of mapping to one of the known synthesis techniques, it is not straightforward how to conceptualize functions that transform emotional significance in terms of conventional parameters usually employed by standard synthesizers.

6.2 Synthesis methods: The background

6.2.1 FM synthesis theory and attributes

Frequency Modulation (or FM) synthesis is a simple and powerful method for creating and controlling complex spectra, introduced by John Chowning of Stanford University around 1973. In its simplest form it involves a sine wave carrier whose instantaneous frequency is varied, i.e. modulated, according to the waveform (assumed here to be another sine wave) of the so-called modulator. This model then is often called simple FM or sine-wave FM. Other forms of FM are extensions of the basic model.

As mentioned before, what is actually varied in frequency modulation is the frequency of a signal, named carrier. Frequency modulation is the method to use in order to produce the vibrato effect:

$$\cos(\omega_c t + f(t)), \tag{6.1}$$

where ω_c the carrier frequency and $f(t)$ a function that is added to the phase of the carrier, thus it ‘modulates” the angle passed to the cosine. Since we can always translate from instantaneous radial frequency $\omega_c t + f(t)$ evaluated at a particular time to the corresponding frequency(differntuate and divide by 2π), this same formula can be viewed as ”frequency modulation”. Strictly speaking, we are referring to frequency modulation as:

$$\cos(\omega_c t + B\int_0^t f(t)dt) \tag{6.2}$$

As soon as $f(t)$ is not directly proportional to the the modulating signal, but to the derivative of the modulating signal, we call this situation frequency modulation (FM) and not angle or phase modulation (PM).

FM modulation importance is that, except for a technique to produce the vibrato effect, under certain circumstances it results to enrich the sound as defined by the carrier frequency with a spectra of harmonic or inharmonic sounds. It is believed that Chowning, the inventor of the FM synthesis technique, stumbled on FM when he sped up vibrato to the point that it was creating audible sidebands (perceived as a timbral change) rather than faster warbling (perceived as a frequency change).

In order to give a clue for the simple aspects of FM synthesis, lets first assume that the function $f(t)$ in 6.1 is a sinusoid of the form so that 6.1 becomes:

$$FM = \cos(\omega_c t + B \sin \omega_m t), \tag{6.3}$$

where ω_m the modulation frequency and B a scaling factor. From 6.3 we take:

$$FM = \cos \omega_c t \cos(B \sin \omega_m t) - \sin \omega_c t \sin(B \sin \omega_m t) \tag{6.4}$$

The quantities $cos(B \sin \omega_m t)$ and $\sin(B \sin(\omega_m t))$ can be expressed with the help of the Bessel functions. At the end we take:

$$FM = \sum_{n=-\infty}^{\infty} J_n(B) cos(\omega_c + n\omega_m)t \tag{6.5}$$

So we end up with a spectrum made up of a "carrier" at w_c and symmetrically placed sidebands separated by w_m. The amplitudes follow the Bessel functions. Because of the relationship in equation 6.5, its possible to boil the control of FM synthesis down to two crucial values, which are defined as ratios of the pertinent parameters. These values are:

- the *harmonicity ratio*, defined as Fm/Fc; this will determine what frequencies are present in the output tone, and whether the frequencies have an harmonic or inharmonic relationship. An integer values ratio correspond to harmonic sounds.
- the *modulation index*, defined as Am/Fm; this value affects the brightness of the timbre by affecting the relative strength of the partials.

6.2.2 Granular synthesis

Granular synthesis is a basic sound synthesis method that operates on the microsound time scale. It is often based on the same principles as sampling but often includes analog technology. The samples are not used directly however, they are split in small pieces of around 1 to 50 ms (milliseconds) in length, or the synthesized sounds are very short. These small pieces are called grains. Multiple grains may be layered on top of each other all playing at different speed, phase and volume.

The result is no single tone, but a soundscape, often a cloud, that is subject to manipulation in a way unlike any natural sound and also unlike the sounds produced by most other synthesis techniques. By varying the waveform, envelope, duration, spatial position, and density of the grains many different sounds can be produced.

The result is usable as music, sound effects or as raw material for further processing by other synthesis or DSP effects. The range of effects that can be produced include amplitude modulation, time stretching, stereo or multichannel scattering, random reordering, disintegration and morphing.

6.3 mapping strategies implemented for Pogany

6.3.1 Direct Mapping strategies

As we have seen in the chapter describing the middle part of the system, features arising directly from the analysis of the gesture were used to the on-line gesture segmentation procedure. Apart from this utility, we could also see them as continuous parameters that adjust music parameters in the synthesis procedure.

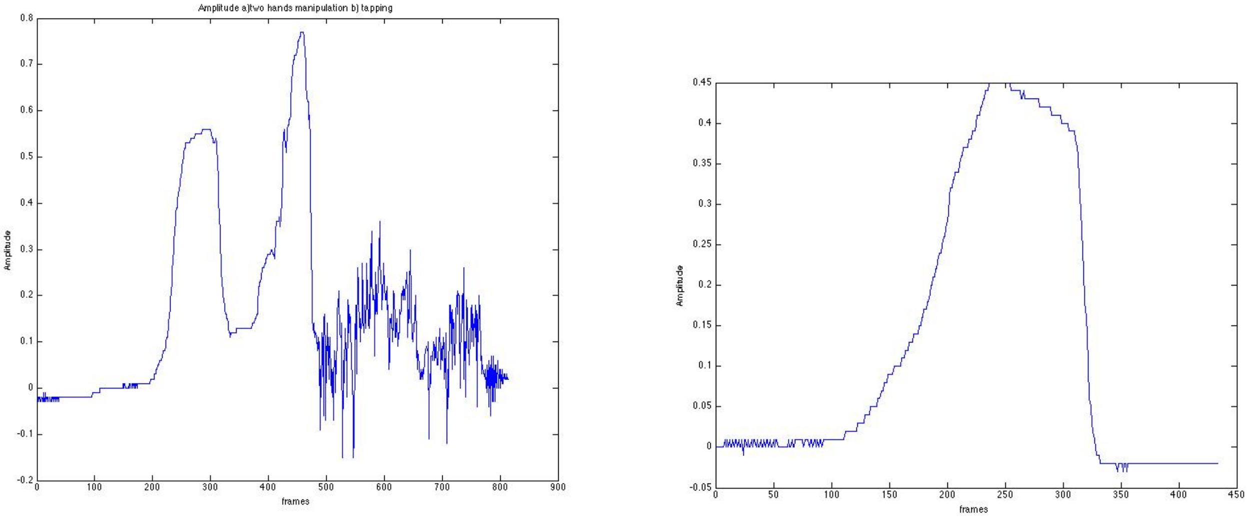


Figure 6.1: On the left: a gesture of tapping on the interface in term of Magnitude per Frame modification. On the right: the gesture of linear transposition, in terms of time, of the hand on zero angle with the face from 20 cm to zero distance

Magnitude Value per Frame and its effect on loudness

The Magnitude Value per Frame represents the mean value of the values of alpha value for each frame(where alpha value is the normalized (1-luminocity) estimate). It is reasonable to think that as the distance of the hand from the activation area of the interface increases, the Magnitude Value per Frame decreases and reversibly. Hence, it seems as a good idea to use this metric in order to model the overall volume of the synthetized sound. In figure 6.1 we can see a number of possible different gestures that horizontally diverge and converge to the interface within the active interface area. As we can see from figure 6.2, the Magnitude Value per Frame becomes zero even from a small distance away from the interface. This would be regarded as non-problematic, even advantageous, in a situation that more weight is given to the capture of haptic gestural activity that triggers sounds. However, it seemed more beneficial to increase the radius of sensitivity around the interface and exploit an larger area of activation. The selection of the transformation of the Magnitude Value per Frame was a conjunction of the need to quasi-linearize the distance factor and to preserve the additive effect of multiple finger haptic interaction in zero distance (figure 6.2). Additionally, in a variation of Magnitude per Frame value for segmented regions of the interface (eyes, mouth etc) that is also used, this modification serves the recovery of mixed haptic and distance activation and to drive this effect to different voices as differential volume schemes. The function selected for this transformation was:

$$F(x) = 1 - e^{-x/a},\ 0 \leq x \leq 1, \tag{6.6}$$

where a a value that controls the gradient of $F(x)$. After trials, the value that fit best our purposes was $a = 1.3 * 10^4$.

The metric for loudness $F(x)$ was used for the control both of the volume of the FM and granular synthesizers. In the case of mapping to more than one voices, a variation of $F_r(x)$, where $r = 1..6$ served as the mixture volume for each of the r corresponding face regions.

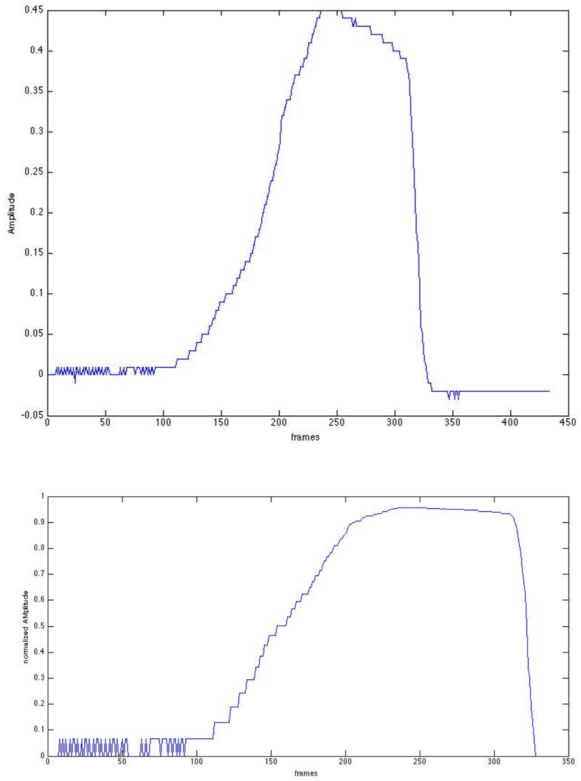


Figure 6.2: A linear transposition in zero angle from the interface, for a trajectory of 20 and 0 cm distance from 'Pogany'. The upper figure represents the Magnitude per Frame value, where the bottom figure the same value after normalization for mapping purposes

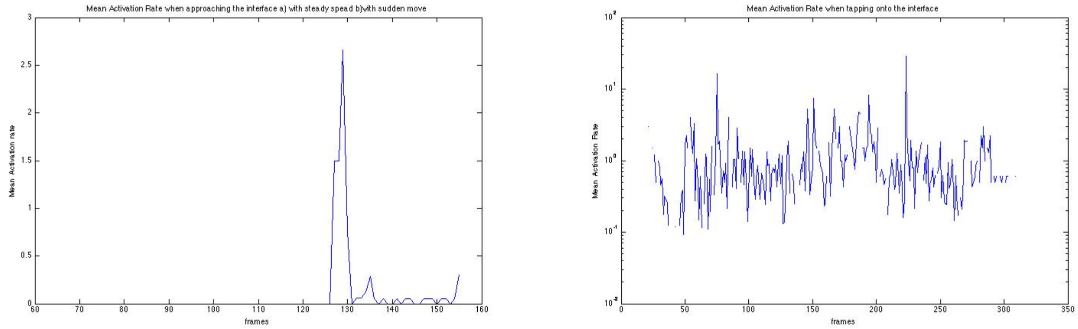


Figure 6.3: Modulation Index. On the left: a non-linear transposition in zero angle from the interface, for a trajectory of 20 and 0 cm distance from ‘Pogany’. Gesture becomes nervous at the ending. On the right: Tapping on the interface

Mean Activation Rate and its correspondence to brightness

Mean Activation Rate was defined in previous chapter as a metric for the speed of the gesture in front the interface. As discussed above, the FM synthesis theory suggests as important factors the modulation index and the frequency ratio. In practice, the modulation index $MI = Am/Fm$ is responsible for the brightness of the sound, as the relative strength of the different sidebands (which affects the timbre) is determined by the relationship between the modulator amplitude (Am) and the modulator frequency (Fm). The physical correspondence under a quickly performed gesture and the brightness of the sound resulted in providing the modulation index with the data from the Mean Activation Rate. Thus, $MAR = MI/b$. Normalization with a factor b was necessary in order to give to the continuously changing value a meaningful -in musical terms- range (6.3).

The MAR metric was respectively adapted to granular synthesis. In order to control the time between grains. Analogously, the time between grains was set to $TMG = c/MIR$, where c a normalization value.

6.3.2 Indirect mapping strategies

At chapter 5 we described how the HMM module is trained and the recognition procedure take place. Recognition acts in two levels of interest for the mapping procedure:

1. the level of gesture recognition. Hand gestures, in terms of intervals of continuous hands activation between two moments of motionlessness, with a number of frames varying from 5-70frames (0.18-2.3 sec), are isolated and probable to get identified by the system. The names of the gesture types and the categorization criteria have a cognitive meaning in terms of face animation, rather than in a face haptics theory. With the help of the visual feedback of the animated head from the screen, the user is urged in an implicit way to assign each of a set of meaningful gestures to a corresponding emotion. For instance, the gesture crossing the eyes area from the down to the upper part, is detected as the ’eyes up’ gesture and causes the head on the screen to animate accordingly, resulting to the emotion ‘surprise’.

2. the level of posture recognition. In between gestures, recognizable or not by the system, whenever hands remain almost steady, the recognizer estimates the probability that this hands

posture corresponds to one of the pre-learned postures. The postures are not assigned to the head animation factors; however, they participate in the mapping functions.

For the mapping procedure we avoided, at first place, getting involved with more complex gestures, although the system architecture permitted such type of output from the gesture recognizer. This was due to mapping restrictions: a complex gestural input, should be decoded with respect to its semantic content, and then adequately linked to a point of an analogous semantic space for the sound timbre. Supposed there is enough evidence about how to map emotions from the head animation to corresponding sounds, there is no enough evidence how to make the gesture-sound correspondences in a case of blended animations in a meaningful and non-arbitrary way.

The decisions we took concerned four principal gestures: ‘eyes up’, ‘eyes close’, ‘smile’, ‘sad’, that correspond to the emotions: ‘surprise’, ‘suspiciousness’, ‘joy’, ‘sadness’ respectively. The attributes of the two synthesis methods we have used have lead us to get advantage of some basic characteristics they provide, translating them to constraints for the mapping procedure.

For this reason, we first tried to set limits for the variation of parameters at FM synthesis. Given that general amplitude and modulation index were driven from direct analysis continuous values, it seemed interesting to experiment with the use of the frequency ratio. From FM theory, the frequency ratio produce harmonic sounds when it is a multiple of 1; on the contrary, when it is for a decimal value of frequency ratio, inharmonic partials become more prominent. In our situation, facial emotions with a rather positive impact, such as joy and surprise, were probable of getting values of harmonicity ratio that lead to a consonant result, in our case, an harmonic sound. From the other side, ‘suspiciousness’ and ‘sadness’, as emotions with mostly negative impact, were less likely to result to an harmonic spectrum. We can find in the bibliography a variety of different approaches for FM synthesis:based on its mathematical background, these approaches aim to a more successful overall control over the synthesis procedure. An interesting theory, directly bounded with practice, is the one by composer Barry Truax [62], which discriminates the sounds according to the frequency ratio family they belong. These families are built as ratios that are assigned to the same normal form and members of each family share common musical attributes. Thus, controlling the frequency ratio, one can gain control over the consonance of the resulting sound.

For the granular synthesis, things are supposed to be easier. The nature of this synthesis itself, as a technique of replaying grains the one after the other helps to emphasize on the timbral aspects of the sound to product, thus creating -in common terminology- soundscapes. With appropriate selections of the audio material used for the grains we can adequately control the nature of the sound to be representative of emotions. In order to correspond the emotions related to the face animation, we made a proper selection of the sound material according to psychoacoustic criteria. In this way, sound samples corresponding to the four basic emotions mentioned before were put to feed the granular synthesizer. The parameters of the granular synthesis affected were the upper and bottom limits of the grain duration, the limits of transposition, the time between grains and of course, the location where the grains were extracted from. The importance of the last is underlined, as each grain source’s correspondence with particular sounds can be evaluated under psychoacoustic criteria.

6.3.3 Environment

The mapping was implemented within the Max environment. Constraint environment, where needed, was helped by externals from the Fuzzy Logic toolbox as well as a number of external libraries (CNMAT, RTCLib). The real-time communication took place between two machines: the first for the real-time HMM recognition, feature extraction and the real-time face animation through the VirChor environment running in Linux operating system and the second for mapping and real-time DSP processing in a MacOSX. The communication between the two machines was established under the OSC (UDP based) protocol.

Chapter 7

Evaluation of ‘Pogany’ as an interface for sound interaction

The previous chapters described the methodology followed in order to provide ’Pogany’ with a gesture recognition mechanism as well as a mapping mechanism to both FM and granular synthesis modules. In order to evaluate our design selections for the interface, we organized an experiment with human subjects that interact musically with the interface. Context, preparation and conclusions arising from this experiment are described in the following sections.

7.1 Context

As the first experiment ever made with Pogany interface for music interaction purposes, it was important at first place to get some clue for the appropriateness of ’Pogany’ for such purpose. Furthermore, as mentioned in the 2nd chapter, discordance over the evaluation criteria and methods do not provide a concrete evaluation process to follow. Under these circumstances, we decided to base our evaluation method on the axis set by Wanderley, adapted to the particularity of the ’Pogany’ interfaces.Thus, the interface set-up for the experiment aimed to give clues for four main attributes:

1. time cotrollability, i.e the potential of the interface in time control, modification of the duration of sounds, chronical accuracy, time quantization legitimacy.
2. sound cotrolability, i.e the sound modification efficiency and facility, the range of possible sounds, the physicality of the correspondences between gestures and sound alterations.
3. learnability, which reflects the easiness in learning gesture patterns and their alliances with the corresponding transformation to sounds and repeat these patterns during performance
4. explorability, the possibility given to explore different sounds and also the expectation for further sound possibilities through the instrument.

However, throughout this master thesis work, the expectation of the author for the importance of high-level gestural information in the music synthesis procedure is more than evident. Given this as a major importance terminus, the axis of the experiment was strongly correlated with the decisions concerning the gesture recognition module and the prospect that such decisions could be found beneficial for the ’Pogany’ virtual instrument design. For this reason, the evaluation process was properly adapted in order to provide an objective measure for judging the effect of high-level discrete gestural information to musical expressivity.

7.2 Preparation of the experiment

The experiment was divided in two sessions. Both sessions made use of the same synthesizer modules, both FM synthesis and granular synthesis, and also shared many other common mapping elements. The main difference between sessions was that, in contrast to the first, the second experiment made use of the high-level gestural information captured on the vicinity of the interface in order to control musical parameters related with the modification of the timbre of the sound. In the following, we will try to describe particular attributes of the configuration of each session seperately.

7.2.1 System configuration

As we have seen in the chapter describing the middle part of the system, features arising directly from the analysis of the gesture, as well as the output of the HMM recognition of a set of both gestures and postures can be provided to the mapping tool. In chapter 'Mapping strategies for Pogany', we discussed the general principles we used for both direct and indirect mapping.

First session The first experiment session was 'direct mapping' oriented. This means that we were motivated to create the 'loudness by distance' and 'brightness by gesture rate' correspondences in order to drive both the FM and the granular synthesizer. Apart from this direct-level mapping, we set an arbitrary change in parameters that, according to our configuration described in 6.3.2, should be controlled by the gestures. Hence, as far as FM synthesis is concerned, harmonicity (frequency) ratio was set to change linearly between *arbitrary* limits and with a slow rate. The same strategy was followed for the granular synthesis module, where the grains were each time selected from one among a set of samples, the duration of the grains was chosen each time through a random procedure between a lowest and a highest value. Transposition effect was controlled the same manner and the time interval between grains was controlled directly by the value of the gesture rate.

Second session For the second session we kept exactly the same settings for the direct level mapping. This time the difference was that, although the samples and the limits of the granular parameters remained unchanged, we used the output from the HMM recognition to control the triggering of different configurations between the granular module. The logic of the correspondences that were created was is described in the 6.3.2 section. At this point, it was thought to be useful to enrich the system with a type of visual feedback. For this reason, an animated head was connected with the system in order to execute the emotion commands through the interface, i.e the four meaningful gestures the system has been trained to recognize. In this way, whenever the user performed a meaningful gesture, this information was set to adjust a set of parameters inside the synthesizer, linearly and with a certain amount of delay, according to the criteria set in section 6.3.2. The meaningful gestures were four and corresponded to four basic emotions: joy, sadness, surprise and suspiciousness. Alteration of one of these 4 moods was triggering relative changes in the harmonicity ratio of the FM synthesis and the duration, pitch and grain source of the granular synthesis module. Additionally, five postures were recognized during session, which corresponded in five different types of activation in front of the main areas of the facial interface: the eybrows, the eyes, the cheeks, the mouth and the nose. The activation of such cues was mapped to result to minor -in comparison with the primal gestures- change on the sound. There was no differentiation in left and right activation; the two symmetrical parts were supposed to be equal, and whenever a gesture was detected it did not matter whether it was coming from the left or the right side. In the meantime, the user was given clue for his action through the visual feedback on the monitor screen. The recruitment of such a feedback was necessary in order to ensure that, even implicitly, the user assigns a set of actions to corresponding emotions, in combination of course with the corresponding sound feedback. Furthermore, the global similarity on the sound quality provided for both experiments, as well as their temporal evolution and duration variability, would allow a fair comparison between other parameters that were different between the two sessions.

Before going ahead, it is important to notice that control gestures are set to get identified only under the assumption that hands are in permanent contact with the interface. This decision in the design of the music interface was taken in order to permit sound events due to activation by distance to evolve freely, as if the gesture recognition component did not existed: This could provide the opportunity for simultaneous direct-level sound manipulation and change in the 'emotional timbre' of the sound.

7.2.2 Conditions of the experiment

The system architecture was as described in previous chapters: The 'Pogany' interface was connected with a Linux Pc, through the VirChor framework with the HMM and the direct gesture analysis module implemented in the internal VirChor code. This machine was communicating the Max environment in a second machine running MacOSX.

As far as the light conditions are concerned, the experiment took place under physical light conditions. The light was slightly non-homogenous, therefore a special training session for the HMM was realized before the experiment with subjects. The training samples were 15-20 for each gesture, coming from one user only. No user adaptation algorithm was applied to the HMM recognition module, as from several tries it was not considered to be particular useful for the instance. The experiment took place at LIMSI laboratory, Orsay.

7.3 the experiment

The procedure had like this: The subject received some explanations concerning the devices that he/she should use and the general concept of the experiment. Then he/she was let some seconds to get familiar with the interface by observation and touch, without any kind of feedback. The next step was to be introduced to the procedure: During two sessions, of 5 minutes each, the subject would be let to interact with the interface in every desirable manner. A subject was encouraged to perform quick or slow movements, by distance or touch, in the front or the vicinity of the interface. Alternative modes of action were also proposed, such as tapping, caressing or hitting (slightly), using each hand separately or both hands simultaneously. After the end of the two sessions, the subjects were asked to fill a questionnaire related to the experiment.

Six subjects passed the music experiment with Pogany. Among them, the five were male and one female. Their age varied from 23 to 29 years old. All subjects had used before interfaces connected to a computer. Moreover, two subjects have made use before of an interface for music over five times, three subjects less than five and for one subject it was the first relative experiance. Concerning their music education, 2 subjects claim to have a strong musical background, being active musicians: One of them was familiar with computer music, while the second just had a listener's experience and is

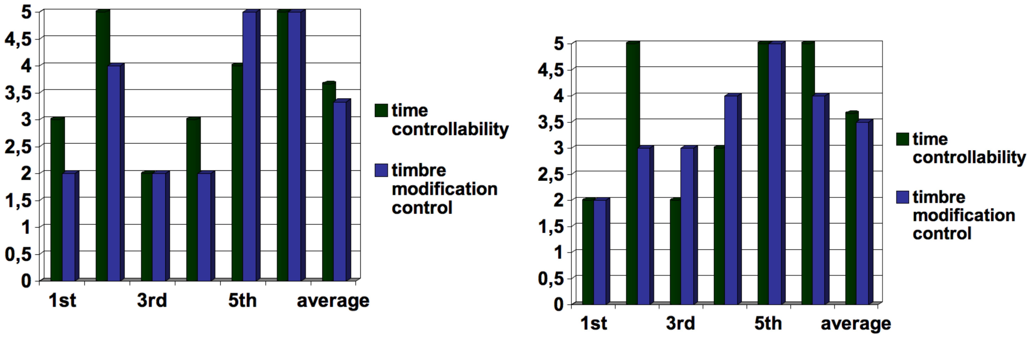


Figure 7.1: Evaluation of the subjects over controllability in time and sound evolution control for the (left) 1st and (right) 2nd session. Horizontal axis represents the subjects, and vertical axis the evaluation score (between 0 and 5)

mostly a traditional musician. For the same question, three subjects claimed to have had a music education of short term in a small age: the interference of the two of them with electronic music was limited and the third one has no experience at all. The sixth subject had had a contact with music only as a listener, and no relation with computer music at all.

After interaction with the interface, all subjects answered to a set of questions concerning their experience. A number of these questions was focussing on eliciting their subjective view for the controllability potential of the setups, in terms of time, sound quality, sound modification capabilities and expressiveness. Other questions concerned the ability for recovering past gesture-sound dyads, and repetition of performed patterns during performance. An other set of questions focused on the easiness to explore new sounds in the given time, and the expectation for an hypothetical second chance. Finally, other questions referred to the visual feedback effect, the willingness of the subjects to experience the same or similar interfaces in the future etc.

Apart from the questionnaires, audio material was collected for each of the sessions of participants, as well as text records concerning the number of meaningful gestures activated for both sessions.

7.3.1 results

Despite the limited number of participants, data gathered proved sufficient to provide important feedback and the base for a number of conclusions. Furthermore, it gave clues for the capability-or not- of the interface for expressive music creation. At first, results will be presented according to the criteria we posed at the beginning of the chapter. Afterwards, an effort will be done to synthesize transversal clues and extract more general conclusions.

In the domain of controllability, the participants found the quality of control in time and timbre adjustment more than satisfying (figure 7.1). In a range of 0 to 5 (with 0 corresponding to ‘very bad’ and 5 to ‘excellent’), subjects evaluated the system with an average of 3.66 and 3.33 for time and timbre modification respectively for the first session. At second session only the average value for the timbre modification increases slightly, while the temporal modification potential remains unchanged. As these parameters are strongly corelated with other factors, as the complexity of mapping and the degree of polyphony, which was not the purpose in our situation, results show the strong acceptance of the interface as a virtual instrument. It is important that the values between the two experiments show minimum differences: this sounds reasonable, as the question mostly referred to the direct mapping strategies that are responsible for the modification of the most prominent parameters of a sound itself: loudness and brightness. These values remain the same throughout the two sessions, a fact that comes to justify the subject's evaluation. An issue to mention is that divergence between the subjects' judgment was large:it is though worth noticing that the the two subjects with some experience in music interfaces evaluated the interfaces with higher notes than the other subjects, and the worst note was put by the subject that had no such experience before.

At this point, it was important to correlate the answers on the questionnaires with real data extracted from the performances. It was difficult to set objective evaluation criteria for the character of each performance. An interesting approach to this matter was to use the normalized mean activation rate (MAR) (see 6.3.1) as a criterion for the kind of activity on the vicinity of the interface: High values of MAR make proof of high velocity in movements. Hence it was decided to calculate the zero crossing rates of $MAR' = m - MAR$, where m the estimated mean value for MAR during gesture activation. Variations of the MAR value for each session and user were recorded. After processing, the mean value of MAR was set to m= 0.7. In 7.1 we show the zero crossing rates for the Transposed MAR (TMAR).

Table 7.1: zero crossing rate of the Transposed Mean Activation Rate

	1st	2nd	3rd	4th	5th	6th	average
TMAR	7	26	27	28	43	69	33.3

Making the comparison of table 7.1 with the figures in 7.1, it is straightforward to understand that the subject who claimed less control had the lowest TMAR score, this means that the amount of general activation rate (i.e the velocity og his gestures) was limited (with a value of 7 to a mean value of 33.3 among subjects). From the other side, subjects 5 and 6 that ranked the system as very good or excellent had a definitely a more 'attacking' approach. is also important to note that this subject is the one that had not any experience of a music interface before, a fact that gives the clue that previous experience with interfaces probably affects the learning curve and the easiness of manipulation, as well the overall view on the effectiveness of such as system.

One of the most important issues of our work concerned the expressivity capabilities through the interface: This fact had a straight effect on decisions taken concerning the configuration of each session. After experiments, participants were asked in the questionnaire to give a judgement about the system expressive capabilities. The question was posed relatively between sessions, asking if there was one session in particular that help them more in expressing themselves. It was impressive that all but one subjects found the second configuration better in expressing themselves, while the other subject found two sessions as of equal expressivity capabilities. This fact, in correlation with the minor divergence in evaluation of time and timbral control between sessions pose an important issue for expressivity related excusively to the process of decoding gesture cues in a higher-level approach employed in the second session. It is also overwhelming that one of the subjects- the one that was statistically found to have the best score of the Transposed Mean Activation Rate value-evaluated the level control as being better in the first experiment, while in the same time confirmed the superiorness of the second session in terms of expressivity.

In terms of learnability, five over the six subjects claimed that they definitely succeed in learning new gestures throughout the little time they were given for manipulation, while the sixth-referred as

1st on statistics- also gave a positive answer but with less certainty. On the question if, even after the experiment, the subject can recall correspondences between gestures and resulting sounds all subject gave a positive answer, each time with more or less certainty. The opinion of the subjects on the matter was of great interest, as with their spontaneous thoughts they have underlined one of the most important issues for an interface: how to establish a learning curve that would not discourage amateurs from getting on with learning and in the same time set high limits for the perfection of performance and thus be intriguing for more experienced users to go on exploring the capabilities of the virtual instrument. Hence, as far as the term of learnability converges with the issues set by the term of explorability, it would be worth having a look at the statements of some of the subjects (1st, 5th and 6th respectively):

'Many difficulties encountered when trying to explore new sounds...difficulties to find a logic and patterns...'

'...For the manipulation some time is necessary to explore the possibilities but when it's done, it is very interesting to produce different sounds.'

'... However, the control on the second experiment was less effective, maybe due to that it demanded a higher degree of expertise gained through practice.' .

In a question asking for the subject's expectation concerning the exploration of new sounds in an hypothetical second chance with the interface all subjects have responded positively, as if the impression created to themselves is that there is still part of the potential of the interface not discovered yet. Some of the subjects underlined the importance of the visual feedback in the form of an animated head for the exloration of the sound capabilities of the interface. Concerning this kind of feedback, all subject found it in all ways useful, also mentioning 'control' and 'logic' in the sound as factors of the creation where it can contribute.

As far as for the general impression from the interface, the one of the 1st subject was rather negative. He insisted in the problems he encountered in trying to understand how exactly it works. From the other part, all other subjects found the interface at least interesting. Several subjects stated that they found the interface innovative in its domain, and one subject suggested its participation on an exhibition with the existing setup as a good idea for the future. Although some subjects claimed not to have familiarity with the 'type' of music it produced, or even not to find it pleasant, this did not prevent them from attaining a good general impression:

'...sometimes it is noisy, but it's funny. I felt like playing (good or bad!) a music instrument...'

A subject underlined the constructive appropriateness of the 'Pogany' interface for such a scope:

'Touching the interface seems important and the contact/touch impression is quite nice...'

Finally, some of the subjects proposed types of usage where setups such as the one of 'Pogany' for music would prove particularly useful, such as for blind people. An inspiring point of view was also set from one of the subjects, mostly concerning intuitive purposes of tangible interfaces for music :

With this interface, people have to guess how to touch it, to learn it by themselves...perhaps a 'traditional' instrument player, after practicing with an interface such as the head interface, will try to find other manners to play with his instrument and produce new sounds.

The impressions we obtained from this experiment were encouraging in many different levels. Firstly, the high-level gestural information decoding module in the second session proved to be particularly useful in terms of expressivity of the user, as stated by all the subjects and confirmed by the equivalence of the two sessions in all other aspects of synthesis' global quality. Second, even through a non-complicated mapping the general impressions for timbre modification and time precision were positive, as well as for the interface itself as a device. Third the interface, except for the first subject that claimed complete unfamiliarity with such interfaces, succeed in providing sufficient conditions for learning patterns and exploring new gestures, with a priority in the advanced users learning curve. Finally, even not proved from the particular experiment, the decisions concerning the expandability options of the setup that were left open during architectural design (such as the option for the interface to be trained for complex gestures) were not discouraged by the results of the experiment.

Chapter 8

Conclusions-Future research

This report signifies the end of first part of research, which aim was to adapt the 'Pogany' affective interface in the framework of music creation and interactivity. For this reason the interface got equipped with an analysis module for the extraction of parameters capable for the drive of continuous musical parameters, as well as with a real-time recognition module for gestures and postures based on the HMM theory and implemented over the speech-oriented HTK library. For the evaluation of the system as an interface for musical expression an experiment was made with real subjects and the results were presented and analyzed.

8.1 General conclusions

Conclusions arising from research and implementations during this Master Thesis can be summarized in three axis: First, the one of the recognition HMM module; second, the usability of such parameters as well as the parameters extracted from direct analysis for a music interface; third, the mapping strategies that should be followed for such a purpose.

Concerning the first, the KeyPoint's alpha values used as an input for the HMM were proved to be sufficient for the recognition of isolated gestures with a high level of accuracy. The training based on Baum-Welsch algorithm, using a reasonable amount of gesture data for each gesture succeeded in building HMM instances with satisfactory discrimination ability. The real-time recognition module got advantage of the low complexity of Viterbi decoding as well as the low-level C and C++ coding framework and it managed to provide accurate estimations with non-critical time, memory and CPU consuming. On the parallel, the token passing algorithm, as a variation of Viterbi decoding, helped in building a framework for the recognition of gesture intentions, based on estimations before the end of each gesture and providing the system with a toolkit that recognizes meaningful gestures from the first moments of their execution. On the parallel, embedded training facilities will allow further enhancements on recognition in the future, giving clues that it will allow the enrichment of the system with a far larger number of recognizable gestures under predefined directional structure definition and grammar constraints.

As far as the usefulness of high-level gestural information for the musical procedure is concerned, the evaluation experiment showed the importance of such information in order to enhance music interfaces such as Pogany with embedded expressiveness. In this scope, multi-modality techniques and multiple types of feedback (haptic, visual) are of major importance for the result.

Finally, the mapping strategies to follow for such a transversal interface architecture are critical; Unless treated with precision, they can deprive such systems of success. However, during the past years, mapping was not treated as a major field of research. This work demonstrated the physical importance of conceptualizing mapping metaphors and revealed the necessity to link both gestural and music data within a common semantic space.

8.2 Future Work

Having a global view over the aspects of interest during this work, it is important to give an idea for the direction of the research to follow. Future work will be presented in two parts: Suggestions for the immediate future and further future plans.

The recognition toolkit performed well during the gesture recognition experiments described in 5.2.2. However, the main problem to resolve is the instability of the system in different light conditions from the ones during training. Despite the amount of trials made concerning the configuration of the HMM recognizer, it would be useful to ensure the optimization of the configuration parameters so that it can work sufficiently under all conditions. Such parameters are the criterion for the convergence of the training algorithm and the amount of gesture data used for training, parameters that must be checked in a global scope in over to avoid overtraining to concretized gesture paths, a particular finger morphology or light environment. Under certain circumstances, spectral features or gradient factors could be useful to apply as an input for the HMM in order to provide the training module with features of greater statistic independence. In would be thus useful to create a database of gesture data from different users and different light conditions and experiment different HMM configurations.

In this light, we can go further with more complex gestures. The aim first is to build a vocabulary of basic gestures-phonemes: regarding them as the structural elements for more complex gestures, we can base the representation of each complex gesture to the concatenation of basic gestures-phonemes. Hence, with the help of a small set of basic gestures, we could build a recognizer for a far larger set of complex gestures. The advantage in our setup for such a procedure is the tools for embedded training the system is equipped with: we can execute embedded training over complex gestures without the need of segmentation, fact that facilitates in a great degree the training procedure. This challenging approach can be further enhanced through other tools borrowed for the HMM technology for speech, such as multiple mixture models configuration and n-gram modelization.

Apart from the previous suggestions for the recognizer, a number of actions in the construction level of the interface could be taken in order to enhance the interface's multimodal sensitivity. The light diffusion inside the interface, which has a definately negative impact on the gesture discrimination capability by the recognizer, could be deteriorated through the use of plastic drivers along the direction of light inserting the holes, as well as the replacement of the existing mirror that reflects light to the camera inside the interface with a more accurate one. As far as the latency problem is concerned, a protocol other than the SVHS could be employed for the connection between the camera and the computer. This suggest the replacement of the existing cheap camera with a Firewire expensive one.

In the application domain, the FFMPEG library used for image processing should be substituted with less consuming algorithms by other processing libraries, as it is the main cause for the latency problems. Except for this, a lot of enhancements could be made inside the code, providing graphical feedback for gestures, a more user-friendly window environment, fully-automated procedures for training. Furthermore, considering the importance of keeping the link between gestures and emotions of the animated head as a visual feedback for the user, new animations should be conceptualized and implemented for each one of the recognizable gestures. Moreover, in order to provide a more complete virtual instrument in a first place, a more complex mapping could be proved fruitful in terms of increasing the variety and complexity of sounds produced and the plurality of the musical result (even through the same methods of synthesis through the employment of multiple voices). As far as the link between gestures and emotions is concerned, a set of emotionally classified sounds through perceptual criteria could certainly enhance emotional impact of the performer when used as an input to the granular synthesizer.

Finally, as far as the mapping domain is concerned, it sounds intriguing to start thinking of conceptualizing more sophisticated approaches to the term 'high-level gestural information'. Even if there is a lot of work done to correlate emotions with the animations of the human face, much less work can be found in literature concerning the relationship between touching a face and conveyed emotions. For an affective interface such as 'Pogany', even if the visual head animation feedback implicitly creates correspondences between users emotions and sound results, a study of relative research in psychology field (such as a model for touching parts of the body) is more than imperative. However, in a second sight, such a model is difficult to evaluate, due to the polyparametric nature of actions of touch among people and the social factor effect. Nevertheless, this would for surely help creating a solid base for the semantic space to which high-level gestural information could be linked.

In the music synthesis domain field, the existing methods of synthesis are not focused on the exploitation of a semantic-level gestural information. Even if FM synthesis and principally granular synthesis can be adapted to such a scope, it is rather doubtful while these methods can succeed in getting advantage of the full potential of a complex gestural semantic space. On the contrary, concatenative music synthesis, in the condition that it gets equipped with the appropriate tools, could maybe confront the difficulties of such a task and be able to drive a sophisticated decoding-synthesis procedure in a high level.

Bibliography

[1] Abowd, G.D. & Mynatt, E.D. (2000) Charting Past, Present, and Future Research in Ubiquitous Computing. ACM ToCHI, Vol. 7, No. 1, pp. 29-58.

[2] Lenman, S., Bretzner, L., Thuresson, B., Computer Vision Based Hand Gesture interfaces for Human-Computer Interaction,CID, Stockholm,Sweden, 2002

[3] Liu, N., Lovell, B., Kootsookos, P., Davis,R., "Model Structure Selection and Training Algorithms for a HMM Gesture Recognition System" Intelligent Real-Time Imaging and Sensing (IRIS) Group, School of Information Technology and Electrical Engineering, University of Queensland, Brisbane, Australia 4072, 2004.

[4] Camurri, A., Mazzarino, B., Menocci, S., Roca, E., Vallone, I., Volpe, G., "Expressive gesture and multimodal interactive systems", InfoMus Lab,-Laboratorio di Informatica Musicale, DIST-University of Genova,2004.

[5] Kurtenbach, G., Hulteen, E. Gestures in Human Computer Communication. In Brenda Laurel (Ed.) The Art and Science of Interface Design, Addison-Wesley, 1990.

[6] Turk, M., Robertson, G. ,"Perceptual User Interfaces", Communications of the ACM, vol. 43, no. 3, 2000

[7] Camurri, A., Mazzarino, B., Volpe, G. Expressive interfaces. Cognition, Technology and Work, Springer-Verlag, December 2003.

[8] Bevilacqua, F., Dobrian, C., Ridenour, J.,"Mapping sound to human gesture:demos from video-based motion capture systems".

[9] Rigoll, G.,Kosmala, A. ,Eickeler, S. "High performance Real-Time Gesture Recognition Using Hidden Markov Models",Duisburg -Germany.

[10] W. M. Newman and R. F. Sproull, Principles of Interactive Computer Graphics! McGraw-Hill, 1979.

[11] Zeller, M., et al.: A Visual Computing Environment for Very Large Scale Biomolecular Mod- eling, Proc. IEEE Int. Conf. on Application-specific Systems, Architectures and Processors (ASAP), Zurich, pp. 3-12,1997.

[12] Berry,G.: Small-wall: A Multimodal Human Computer Intelligent Interaction Test Bed with Applications, Dept. of ECE, University of Il linois at Urbana-Champaign, MS thesis,1998.

[13] .Ju, S., Black, M., Minneman, S., Kimber, D.: Analysis of Gesture and Action in Technical Talks for Video Indexing, IEEE Conf. on Computer Vision and Pattern Recognition, CVPR97, 1997.

[14] .Davis,J., Bobick,A.: Virtual PAT: A Virtual Personal Aerobic Trainer, Proc. Workshop on Perceptual User Interfaces, pp.13-18,1998.

[15] . Quek, F.: Unencumbered Gestural Interaction, IEEE Multimedia, Vol.3, No.4, pp.36-47, 1997.

[16] . Crowley,J., Berard,F., Coutaz,J.: Finger Tracking as An Input Device for Augmented Reality, Int.Workshop on Automatic Face and Gesture Recognition, Zurich, pp.195-200,1995.

[17] . Triesch, J., Malsburg, C.: A Gesture Interface for Human-Robot-Interaction, Intl Conf. On Automatic Face and Gesture Recognition, 1998.

[18] Vatavu, R., Grisoni, L., Pentiuc, S., "Gesture Recognition Based On Elastic Deformation Energies", Laboratoire d'Informatique Fondamentalle de Lille, 7th workshop on gesture in Human Computer Interaction and simulationGW, 2007, Lisboa, Portugal, 2007.

[19] Camurri, A., Canepa, C., Ghisio, S., Volpe, G., "Automatic Classification of Expressive Hand Gestures on Tangible Acoustic Interfaces According to Laban's Theory of Effort', InfoMus Lab, DIst, Genova, Italy, 7th workshop on gesture in Human Computer Interaction and simulationGW, 2007, Lisboa, Portugal, 2007.

[20] . Imagawa, K., Lu, S., Igi, S.: Color-Based Hand Tracking System for Sign Language Recognition, IEEE Int. Conf. on Automatic Face and Gesture Recognition, Japan, 1998.

[21] Hunt, A., Kirk, R., "Trends in Gestural Control of Music", chapter Mapping Strategies for Musical Perfor- mance". Ircam - Centre Pompidou, 2000.

[22] Wanderley, M., Orio, N., Schnell, N. "Towards an Analysis of Interaction in Sound Generating Systems", Proc. of the International Symposium on Electronic Arts - ISEA2000, Paris - France, 2000.

[23] C. Roads. Computer Music Tutorial, chapter Musical Input Devices, pages 617658. The MIT Press, 1996.

[24] Rich, Robert. "Buchla Lightening MIDI Controller." Electronic Musician 7, 10 (1991): 102-108, 1991.

[25] Juslin, P., 'Five Facets of Musical Expression: A Psychologist's Perspective on Music Performance', Uppsala University, Psychology of Music, Vol. 31, No. 3, 273-302, 2003.

[26] Cooper, Douglas. "Very Nervous System." Wire Magazine. 3, 3 (1995): 134-170, 1995.

[27] Lovell, Robb and John D. Mitchell. "Using Human Movement to Control Activities in Theatrical Environments.", Proceedings for the Fifth Biennial Symposium for Arts and Technology, ed., N. Zahler. New London: Connecticut College, 1995.

[28] Coniglio, M., "In Plane". Personal conversation. 1995.

[29] Knapp, B. ,Lusted, H.. A Bioelectrical Controller for Computer Music Applications. Computer Music Journal, 14(1):p.42, 1990.

[30] Waiswicz, M. "THE HANDS: A Set of Remote MIDI Controllers.", Proceedings of the 1985 International Computer Music Conference, ed., B. Truax. San Francisco: Computer Music Association, 1985

[31] Tanaka, A., Knapp, R., "Multimodal Interaction in Muisc Using the Electromyogram and Relateive Position Sensing", 2002.

[32] Miranda, E., Gimenes, M., "Improvisation for Two Pianos and Brain-Computer Music Interface", workshop for computer improvisation, NIME 06, Paris, 2006.

[33] Hasan L., Yu, N., Paradiso, J., "The Termenova: A Hybrid Free-Gesture Interface", Responsive Environments Group MIT Laboratory, NIME- 02, Dublin, 2002.

[34] , Study of haptic and visual interaction for sound and music control in the Phase project, Rodet, X., Lambert, J., Cahen, R., Gaudy, T., Guedy, F., (IRCAM), Gosselin, F., (CEA-LIST), Mobuchon, P., (ONDIM), Proceedings of NIME 05 conference, Vancouver, Canada, 2005.

[35] Modler, P., Myatt, T., 'Image Features Based on Two-dimensional FFT for Gesture Analysis and Recognition', SMC07, Leykada, Greece, 2007.

[36] Pritchard, B., Fels, S., "GRASSP: Gesturally-Realized Audio, Speech and Song Performance", NIME 06, Paris, France, 2006.

[37] Weinberg, G., Thatcher, T., Gatech, "Interactive Sonification of Neural Activity", NIME 06, Paris, France, 2006.

[38] Rovan, J., Wanderley, M., Dubnov, S., Depalle, P., "Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance", Analysis-Synthesis Team/Real-Time Systems Group- IRCAM-France, Journees d'Infromatique Musicale, JIM97, 1998.

[39] Jord, S., "a. Digital instruments and players: Part I efficiency and apprenticeship", Proceedings of the 2004 Conference on New Instruments for Musical Expression (NIME-04), 2004.

[40] Blaine, T., Fels, S., "Contexts of collaborative musical experiences", Proceedings of the 2003 Conference on New Instruments for Musical Expression NIME-03, 2003.

[41] Wanderley, M., Orio, N. "Evaluation of input devices for musical expression; borrowing tools from HCI", Computer Music Journal, 26(3):6276, 2002.

[42] D. Isaacs. "Evaluating Input Devices for Musical Expression", Thesis 2003, http://www.itee.uq.edu.au/ markp/publications/ DavidIsaacsThesis.pdf, 2003.

[43] Juslin, P. "Music and Emotion, chapter Communicating Emotion in Music Performance: A Review and Theoretical Framework", pages 309337. Oxford Univ. Press, 2001.

[44] Chew, E., Franois,A., Liu, J., Yang, A., "ESP: A Driving Interface for Expression Synthesis" University of Southern California, Viterbi School of Engineering, NIME05, Canada, 2005.

[45] Machover, T., "Harmonic Driving", brainop.media.mit.edu/text-site/onsite/harmony.html.

[46] Yonezawa, T., Suzuki, N., Mase, K., Kogure, K., "HandySinger: Expressive Singing Voice Morphing using Personified Hand-puppet Interface " , NIME 06, Paris, 2006.

[47] Bevilacqua, F., Rasamimanana, N., Flety, E., Lemouton, S., Baschet, F., ”The Augmented Violin Project: research, composition and performance report”, NIME06, Paris, 2006.

[48] Cont, A., Coduys, T., Henry, C., ’Augmented Mapping: Towards an intelligent user- defined gesture mapping’, SMC04, 2004.

[49] Cont, A., Coduys, T., Henry, C., ’Real-time Gesture Mapping in Pd Environment using Neural Networks’, NIME04, 2004.

[50] Cui,Y, Weng,J.: Hand Sign Recognition from Intensity Image Sequences with Complex Background, Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.88-93, 1996.

[51] Polana, R. Nelson, R.: Low Level Recognition of Human Motion, IEEE Workshop on Motion of Non-Rigid and Articulated Objects, Austin, pp77-82, 1994.

[52] Watanabe, T., Yachida, M.: Real Time Gesture Recognition Using Eigenspace from Multi Input Image Sequences, Intl Conf. On Automatic Face and Gesture Recognition , Japan, 1998.

[53] L. Rabiner, A tutorial on Hidden Markov Models and selected applications in Speech Recognition, Proceedings IEEE, pp. 257-284, February 1989.

[54] Jacquemin, ”C., Pogany: A Tangible Cephalomorphic Interface for Expressive Facial Animation”,LIMSI-CNRS and Univ. Paris 11,Orsay France, 2007

[55] htk.eng.cam.ac.uk/

[56] Fagerberg, P.,Stahl, A., Hook, K., Designing Gestures for Affective Input: An Analysis of Shape, Effort and Valence, Stockholm University/KTH.

[57] Sundstrom, P., Stahl, A., Hook, K.” In situ informants exploring an emotional mobile meassaging system in their everyday practice”. Int. J. Hum.-Comput. Stud. 65(4) (2007) 388403 Special issue of IJHCS on Evaluating Affective Interfaces.

[58] Jie Yang, Yangsheng Xu ”Hidden Markov Model for Gesture Recognition”, report CMU-RI-TR-94-10, The Robotics Institute Carnegie Mellon University Pittsburgh, Pennsylvania 15213, May 1994.

[59] Paiva, A., Andersson, G., Hook, K., Mourao, D., Costa, M., Martinho, C.” Sentoy in FantasyA: Designing an affective sympathetic interface to a computer game.” Personal Ubiquitous Comput. 6(5-6) (2002) 378389

[60] Ekman, P., Friesen, W.V.: Facial action coding system: A technique for the measurement of facial movement. Consulting Psychologists Press, Palo Alto, CA, USA,1978.

[61] Jacquemin, C., Virtual Choreographer, virchor.sourceforge.net/.

[62] Truax, Barry, FM tutorial, http://www.sfu.ca/ truax/fmtut.html

[63] Maniatakos, F., ’Affective interface for emotion-based music synthesis’, Sound and Music Computing conference SMC07, Leykada, Greece, 2007.