Building Fuzzy Rules in an Emotion Detector

Alain Bonardi
MSH-Paris Nord
UMS 2553
4, rue de la Croix Faron
Plaine Saint-Denis
93210 SAINT-DENIS
alain.bonardi@univ-paris8.fr

Isis Truck
MSH-Paris Nord
UMS 2553
4, rue de la Croix Faron
Plaine Saint-Denis
93210 SAINT-DENIS
truck@ai.univ-paris8.fr

Herman Akdag
LIP6 – UPMC
UMR 7606
8, rue du Capitaine Scott
75015 PARIS
Herman.Akdag@lip6.fr

Abstract

In this article, we propose a methodology to automatically construct fuzzy rules in a Fuzzy Rule-Based System that detects emotions from an actor's performance during a show.

The idea is to collect video data from a lot of performances of the same show from which it should be possible to visualize all the emotions and intents or more precisely "intent graphs". To do this, we analyze the collected data defining low-level descriptors that are converted into high-level characterizations thanks to aggregations. The following step is to partition correctly the universes in classes, depending on the retrieved data. Finally the fuzzy rules built from the classes are used to label the detected emotions.

Keywords: Assistant, Fuzzy Rules, Intents, Emotions, Detection, Art Director, Performance.

1 Introduction

Entertainment productions are more and more numerous and directing is often a complex task, notably with the productions containing a part of improvisation. However the performers of these productions need to be directed since the improvisation often follows some rules.

In this context, it is necessary to conceive tools that could help the art director in his task of actors' performance supervision. The tool we propose here is a kind of assistant that gives a representation of a set of complex data for the exploration and observation of a show, and it is also a way of better understanding the creation and execution of the show. One important point is that, as every computer program, the assistant must be deterministic and systematic, i.e. it should always give the same results for a given entry and it must look over the whole data. Like high-level sportsmen that have tools to analyze, correct and improve their gesture, we want to propose a tool to assist the creative artists, to let them better understand the phases of a performance and to help them in their creation process.

We have chosen to work on a virtual opera called *Alma Sola* written by Bonardi and Zeppenfeld where a performer plays (sings and dances) different blocks from different *universes* (such as Prologue, Love, Pleasure, etc.). *Alma Sola* is an open form of opera [1]. The performer embodies a feminine Faust and wanders through the various *universes* split into blocks. She therefore interprets an opera playlist that she selects during the show itself. For instance, a performance can be : Love-3, then Wealth-5, then Pleasure-3, etc. The computer offers continuations to the performer and suggests the next block to be performed (cf. figure 1).

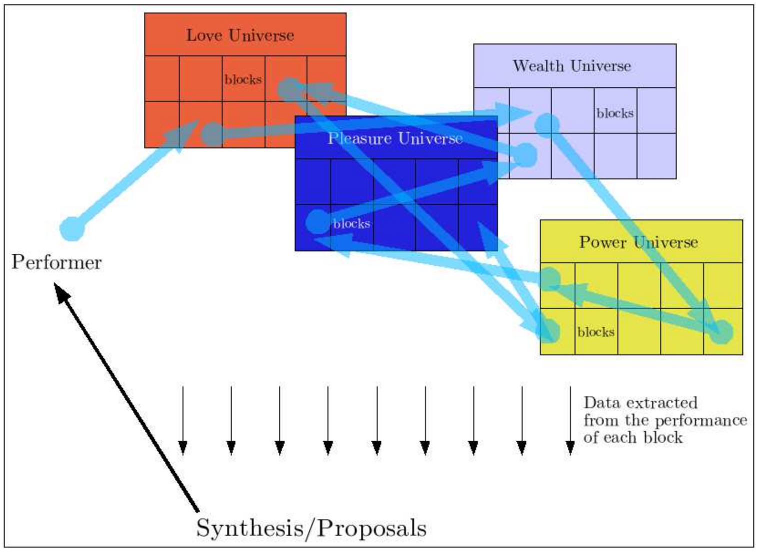

Figure 1: *Alma Sola* sequence diagram.

We retrieve the actor's performance on video files and look for emotions in the data. A distinction must be made between intent and emotion: indeed intent corresponds to the conscious part of the emotion. Thus the assistant enables the comparison between the performer's intent and the rendered emotions.

It is widely known that fuzzy logic offers good tools to deal with such subjective concepts [10], emotions here, this is why we shall use a fuzzy rule-based system (FRBS) to detect the performer's emotions.

The article is organized as follows: first we give an overview of the existing research about assistants in art, then we describe our assistant and notably the way we retrieve emotion descriptors. In the third section we show how the descriptors are partitioned in classes. This step is necessary to build correctly the fuzzy rules that form the inference system. Finally, section 4 concludes this study.

2 Research about performer's assistants

In a way, research about performer's assistants has existed for centuries. One immediately thinks of the use of mirrors in dance. At the time when mirrors become common place in Europe (Renaissance), the first treatise about dance is released by Thoinot Arbeau in 1589. Dance has given the example for a long time among performing arts. Many notation systems (Feuillet, Laban, Benesh, etc.) have been developed along centuries. All these systems, as the musical notation, have in common to give through a score the causes of the phenomenon (and not its result) to try to access the author's intentions. For instance, scores encode descriptions either implicit (you have to play an F, but the score does not tell you how to do it) or explicit gestures to be achieved (cross hands when playing the piano, for instance), and from these structured indications, a dancer or a musician tries to infer some of the author's intentions [5].

Another approach consists in dealing with another set of intentions, the ones expressed by the performers such as we experience them, without direct reference to the author's. Hopefully these two systems of intention have a non-empty intersection (we sometimes feel this is not the case or on the contrary we sometimes feel a perfect adequacy between the work and its performers). In the field of music, for instance, one can "measure" a kind of difference between the prescription (typically the score) and its realization (from listening to musicians). In his approach of "Virtual scores", Manoury has considered [7] in his pieces for solo instrument and live electronics the computation of this difference (*Jupiter* for solo flute and live electronics, 1987, *En Echo* for voice and live electronics, 1991).

The first step in that approach consists in measuring various aspects of the performance, using captors in a broad meaning [11]. Various captors are nowadays available: video camera, wireless microphone, ultrasound device, carpet detectors, digital compass, etc. Recently, Camurri and his team have developed the first robust platform for the analysis of gestures and consequently of performer's emotions. It is named EyesWeb [3], [6]. It is not based on captors implemented on the performer's body (with heavy batteries and radio transmission), but on video capture with a static shot. It is based on a graphical language that implements many descriptors of gestures: quantity of motion, stability, etc. EyesWeb has become a worldwide standard for performance analysis.

Roughly ten years before, the Ircam institute (and Cycling 74 company) had developed the real-time digital sound analysis and synthesis platform to complete Max software. It is named MSP (Methods for Sound Processing) and is now included in Max software (Max/MSP). This software is a worldwide standard in real-time sound analysis.

At the present time the state-of-the-art consists mainly in inferring a few emotional states from the raw data delivered by EyesWeb and/or Max/MSP. A project relatively similar to ours is developed by Friberg and his team. They have conceived a real-time algorithm to analyze emotional expression in musical performance and body movement [4]. In the framework of a game named "Ghost in the cave", the player has to express different emotions using his/her body or his/her voice, and these emotions are the input values of the software. They use EyesWeb to recover body movements and sound descriptors (sound level, instant tempo, articulation, attack rate, high-frequency content). But in this game, there is an immediate feedback, and the player has to move constantly and talk until the software reacts according to his/her wishes. Our aim is not the same, i.e. the assistant adapts to the performer and not the contrary. There are also many works on facial expressions recognition with fuzzy techniques, such as the approach developed by Ralescu & Hartani, for example [9].

3 Project Description

As explained above, we have chosen to work on the interactive opera *Alma Sola* written by A. Bonardi and C. Zeppenfeld. Two dissimilar scenes have been extracted to be used as "sample scenes": they are the chanted Prologue (which is improvised, unwritten) and the Love Universe (which is "strictly written"). The performer (a singer/dancer, here) is filmed by a camera in wide and static shot (cf. figure 2, left) and his/her voice is recorded in a separate file, in order to handle both sources of data separately.

The project is centered on two main phases: the acquired scene processing (with the performer) and the sound capture processing. In this article we focus on the first phase, where EyesWeb has been used (through a dedicated patch we have written) to analyze accurately, understand and exploit non-verbal expressive gestures. Several parameters can be extracted from the video file: quantity of motion, stability, motion duration, pause duration in a scene, contraction index and surface of the performer in the image, convex hull of the body silhouette, velocity, acceleration, etc.

This patch construction step is not trivial and represents hours of tests with sample videos and it implies a concertation with the art directors. First of all, the background of the image must be removed — using the difference between two frames — in order to keep only the performer's movement, since the camera is static. Next, the parameters can be easily extracted because we are sure they only concern the performer, and not the background.

Here is the description of some of the most interesting parameters for our problem (Table 1 shows some of them). These parameters can be seen as gesture primitives.

The quantity of motion is computed by the

Figure 2: Left, an image taken from the video file; right, the corresponding convex hull of the body silhouette (performer: Claire Maupetit).

Descriptor	**Interval**	**Intuitive description**
Movement duration	$[0, +\infty[$ in ms	
Pause duration	$[0, +\infty[$ in ms	
Contraction surface	[0,nb of pixels in the video[	Quantity of pixels representing the performer in the image
Contraction index	$[0, 1]$	Large: performer's stance is open Medium: performer's stance is normal Small: performer's stance is closed
Contraction matrix coord.	$x, y \in$ [0,nb of pixels in the video[	Performer location in the image
Stability	$[0, 2.34]$	Low: performer is close to the ground, legs spread High: performer is standing, legs tight
Quantity of movement	$[0, 1]$ 0: no movement; 1: all parts of the silhouette have moved	Speed of movement and displaced mass
Center of gravity	$x, y \in$ [0,nb of pixels in the video[	

Table 1: Descriptor definitions.

number of pixels changing position between two instants (the *white pixels* in figure 2); the convex hull of the body silhouette is the bounding rectangle of the *white pixels* (cf. figure 2); the stability is the ratio of the height of the silhouette's center of gravity on the length of the segment connecting the lower points of the silhouette; the contraction index is the ratio of the silhouette's surface over the surface of the convex hull. The stability is an important descriptor since it gives good insight on whether the performer is near the ground or not, i.e. whether the performer puts himself at risk or not. The contraction index is also very useful and reflects whether the performer is effusive or not.

Choosing these parameters judiciously (called video descriptors) allows us to compute various aggregations of each set of values for each descriptor (cf. a simple example in figure 3).

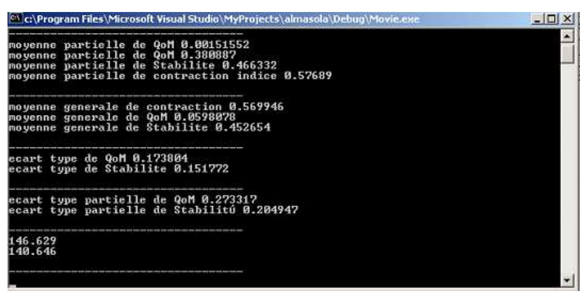

Figure 3: Some of the aggregations used.

The chosen aggregators are: partial and general means (computed on a part of the captured scene or on the whole scene), standard deviation, covariance, etc. denoted $A_1, A_2 \ldots A_9$. Then, to a meta-level, we characterize and categorize the sequences of each scene thanks to an FRBS. The number of categories used in this opera is five (Sleepy, Angry, Happy, LoveBeliever (could be also called

Effusive), LittleEffusive). Figure 4 sums the whole process up.

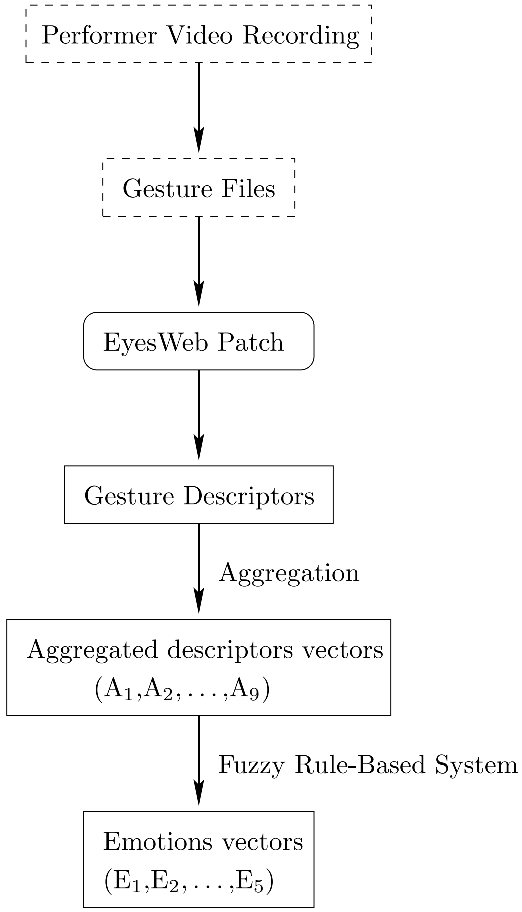

Figure 4: Our system components.

(We capture 25 frames per second, i.e., 25 values per second for each gesture descriptor.)

As can be seen in Figure 4, once the descriptors are extracted from the video files, they are aggregated in order to give a description of each emotion we want to recognize. Figure 5 shows graphically the results for the Universe of Love. The y-axis represents the normalized (between 0 and 2) interval described in table 1, i.e. it is in ms, number of pixels or ...

The next step is to classify in partitions the aggregation results. For example, when the performer expresses an emotion such as happiness, he/she moves a lot. But this is not always easy to guess even if the emotions share some general patterns.

The fuzzy partitioning allowing the construction of the rules is now described.

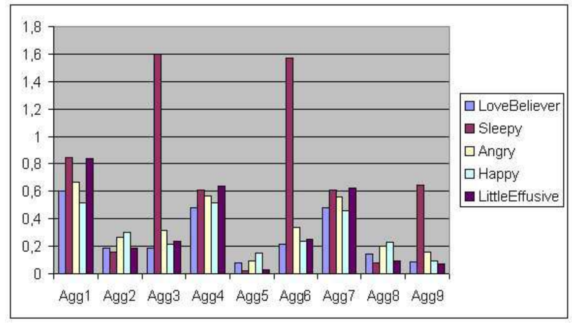

Figure 5: Aggregated descriptors for each emotion in the Universe of Love.

4 Building the Fuzzy Rules

Fuzzy partitioning of the values taken by the aggregation results must cover the domain of discourse. However a simple uniform distribution of the classes on the axis is not appropriate since sometimes small, other times average or in other cases big variations lead to an emotion change, depending on the aggregated vectors.

Like Martínez & al. in [8], we propose a categorization depending on the data distribution. First of all, five classes are considered: Very Low, Low, Average, High and Very High values (denoted *VL*, *L*, *A*, *H*, *VH*). For the moment, we consider the classes as simple intervals. The fuzzy sets construction will be explained below. For each aggregator and for all the records of each scene, the building of class Low is dynamically performed as follows: the interval's left bound is the minimum value of the whole data denoted x_0. This way, when analyzing a new record of the same scene, if there are smaller values than x_0, they will be categorized as Very Low values. In the same way, for each aggregator and for all the records of each scene, the building of class High is dynamically performed as follows: the interval's right bound is the maximum value of the whole data denoted x_2. Then, for each aggregator and for all the records of each scene, the average value of the whole data is denoted x_1. Both intervals $[x_0, x_1]$ and $[x_1, x_2]$ are split into three equal sub-intervals, each.

Finally, the interval for class L is the concatenation of the first two sub-intervals, the inter-

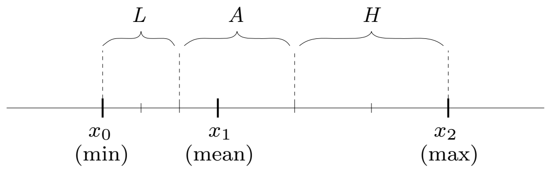

Figure 6: Splitting the intervals.

Emotions	$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{A}_3$	$\mathbf{A}_4$	$\mathbf{A}_5$	$\mathbf{A}_6$	$\mathbf{A}_7$	$\mathbf{A}_8$	$\mathbf{A}_9$
LoveBel.	*L*	*A*	*L*	*L*	*A*	*L*	*L*	*A*	*L*
Sleepy	*H*	*L*	*H*	*H*	*L*	*H*	*H*	*L*	*H*
Angry	*A*	*H*	*L*	*A*	*A*	*L*	*A*	*H*	*L*
Happy	*L*	*H*	*L*	*L*	*H*	*L*	*L*	*H*	*L*
LittleEff.	*H*	*L*	*L*	*H*	*L*	*L*	*H*	*L*	*L*

Table 2: Classification for the Universe of Love.

val for class A is the concatenation of the next two sub-intervals and the interval for class H is the concatenation of the last two sub-intervals. Figure 6 shows the intervals split. Thus it is possible to classify the aggregation values for the five emotions, as shown in table 2 where A_i is the i^{th} aggregator, in the case of the Love Universe. This will permit to establish the fuzzy rules.

The next step consists in building the fuzzy subsets representing the five classes for the FRBS. Figure 7 shows how this is performed. The smallest class-interval, L in our example, is defined as an L-R fuzzy number and sets the construction unit for the other intervals.

If we denote by (a_F, b_F, c_F, d_F) an L-R fuzzy subset F (with $\mathrm{Supp}(F) = [a_F, d_F]$ and $\mathrm{Ker}(F) = [b_F, c_F]$) [2], the building of the five classes in the example of figure 7 is as follows:

$$VL \;:\; \begin{cases} a_{VL} = b_{VL} = \inf_{x \in A_i} x \\ c_{VL} = 2x_0 - x_{0,1} \\ d_{VL} = x_{0,1} \end{cases}$$

$$L \;:\; \begin{cases} a_L = 2x_0 - x_{0,1} = c_{VL} \\ b_L = c_L = x_{0,1} = d_{VL} \\ d_L = x_1 \end{cases}$$

$$A \;:\; \begin{cases} a_A = x_{0,1} = b_L \\ b_A = x_1 = d_L \\ c_A = x_{0,2} + x_{1,1} - x_1 \\ d_A = 2x_{1,1} - c_A \end{cases}$$

$$H \;:\; \begin{cases} a_H = c_A \\ b_H = d_A \\ c_H = 2x_{1,2} - b_H \\ d_H = 2x_2 - c_H \end{cases}$$

$$VH \;:\; \begin{cases} a_{VH} = c_H \\ b_{VH} = d_H \\ c_{VH} = d_{VH} = \sup_{x \in A_i} x \end{cases}$$

This way, the smallest class-interval, either L or H, is an L-R fuzzy number. Generally speaking, at least, one of the two classes (L or H) must be an L-R fuzzy number. Nevertheless A and H (or respectively A and L, depending on which class contains the smallest sub-interval) may be either L-R fuzzy numbers or L-R fuzzy intervals. They are actually fuzzy numbers under a certain condition depending on a parameter that we call ε:

$\forall X \in \{L, A, H\}$, we denote by X^p the preceding class in the universe. After having computed a_X, b_X, c_X, d_X such as in the example of figure 7 for class A or H, if $c_X - b_X < \varepsilon$ then c_X and b_X must be re-computed, i.e. $b'_X = c'_X = 1/2(b_X + c_X)$

The value taken by ε determines the membership value v to which the classes overlap.

In particular, if $\varepsilon < c_X - b_X$ then $v = 0.5$ and X is an L-R fuzzy interval, otherwise (i.e. $\varepsilon \geq c_X - b_X$) $v = \dfrac{1}{a_X - b'_X}\left(\dfrac{a_X c_{X^p} - b'_X d_{X^p}}{-a_X + b'_X - c_{X^p} + d_{X^p}} + a_X\right)$ and X is an L-R fuzzy number.

In the software we propose (see section 5), we also offer the possibility to use L-R fuzzy numbers exclusively (cf. figure 8) or L-R fuzzy intervals exclusively.

Finally, it is easy to establish the fuzzy rules according to Table 2, one rule per line. Here is an example for the Love Universe:

```
Rule 1:

If A1 is L & A2 is A & A3 is L &
   A4 is L & A5 is A & A6 is L &
   A7 is A & A8 is L & A9 is L
Then the emotion is LoveBeliever
```

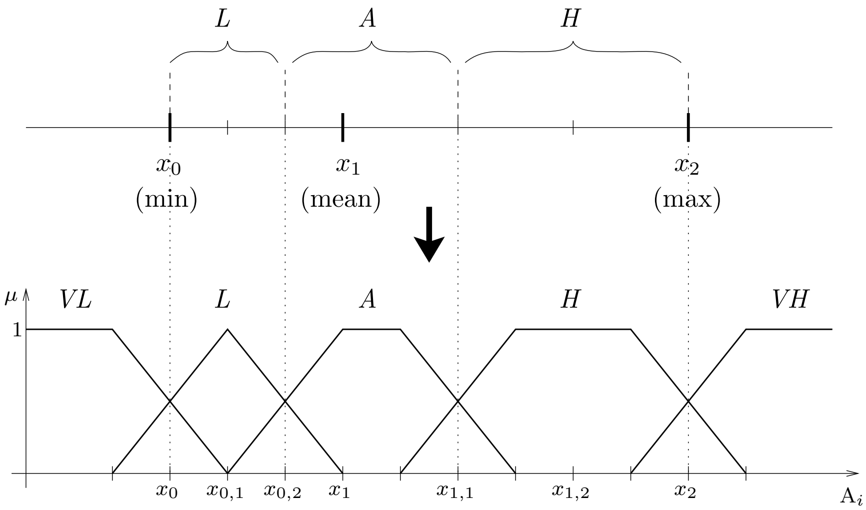

Figure 7: Building the fuzzy classes.

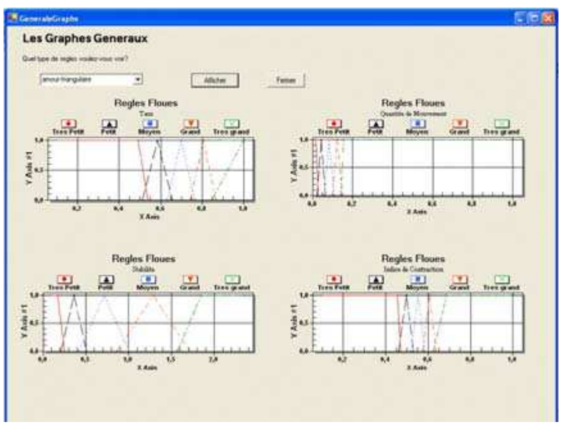

Figure 8: The L-R fuzzy numbers used in the software.

5 Application

The application we have developed implements the concepts explained above. We have worked on two universes: Love and Prologue. The performer has played several times each universe and the software has tried each time to detect the emotions perceived. The video files obtained for the performances are split into smaller files that are given to the software. After having chosen the fuzzy subsets that will be used for the partitioning (cf. section 4), the values of the aggregated descriptors are displayed and a graphical result is also proposed. Figure 9 shows that both Love Universe and Prologue have rather been performed with Sleepy emotion.

The results are better (i.e., the results for the emotions are more clearcut) using partitioning with both L-R fuzzy numbers and L-R fuzzy intervals because they adapt to the level of precision, depending on the values retrieved from the video file.

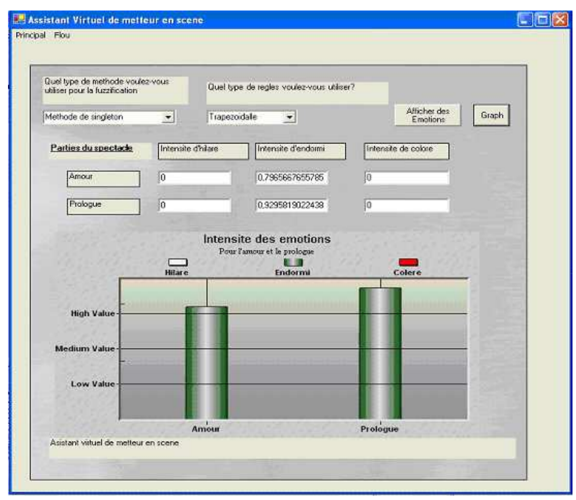

Figure 9: A screenshot of the software.

6 Conclusion

In this paper we have presented a system that is able to give clues to the art director in order to evaluate a performer's rendition. This is done thanks to a fuzzy rule-based system that detects the actor's emotions during a performance show. One originality is the way we construct the fuzzy classes when partitioning the universes before the rule construction. They are dynamically built according to the values characterizing the performance.

As a future work, we will evaluate our system with musical experts by asking them to fill in a perception grid for each scene. Moreover, we will try our assistant on other test sets, i.e. with more records from *Alma Sola* (with the same performer or not) but also with records from other shows. Moreover, it would be very interesting to include a back propagation in the software: when an unexpected emotion is detected, the assistant should suggest modifications of his behaviour to the performer in order to obtain best results during the next detection.

Concerning *Alma Sola* opera itself, we can also imagine in the future that the assistant could be used to classify the blocks performed according to the detected emotions and then contribute to the design of the open form.

Acknowledgements

We thank Adrien Revault d'Allonnes and Murat Goksedef for their collaboration and help on this work.

This project was funded by the Maison des Sciences de l'Homme Paris Nord.

References

[1] A. Bonardi & F. Rousseaux. New Approaches of Theatre and Opera Directly Inspired by Interactive Data-Mining. In *Proceedings of the Int. Conf. Sound & Music Computing (SMC'04)*, pages 1-4, Paris, France, 2004.

[2] B. Bouchon–Meunier. La logique floue et ses applications. Addison–Wesley, 1995.

[3] A. Camurri, M. Ricchetti & R. Trocca. EyesWeb - toward gesture and affect recognition in dance/music interactive systems. In *Proceedings of the Int. Conf. IEEE Multimedia Systems*, Firenze, Italy, 1999.

[4] A. Friberg. A fuzzy analyzer of emotional expression in music performance and body motion. In *Proceedings of Music and Music Science*, Stockholm, 2004.

[5] K. Kahol, P. Tripathi & S. Panchanathan. Automated Gesture Segmentation From Dance Sequences. In *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR 2004)*, pages 883-888, Korea, 2004.

[6] E. Lindström, A. Camurri, A. Friberg, G. Volpe & M.-L. Rinman. Affect, attitude and evaluation of multi-sensory performances. In *Journal of New Music Research*, 34(1): 69-86, Taylor & Francis, 2005.

[7] P. Manoury & M. Battier. Les partitions virtuelles. In *Ircam documentation*, Paris, 1987.

[8] L. Martínez, J. Liu, Da Ruan & J.B. Yang. Fuzzy Tools to Deal with Heterogeneous Information in Engineering Evaluation Processes. In *Information Sciences*, In Press, 2006.

[9] A. Ralescu & R. Hartani. Fuzzy Modeling based Approach to Facial Expressions Understanding. In *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 1(1), pages 45-61, 1997.

[10] L.A. Zadeh. Toward a generalized theory of uncertainty (GTU) — an outline. In *Information Sciences*, Elsevier, 172 (1-2), pages 1-40, 2005.

[11] C. Zeppenfeld. L'acteur face aux technologies, Master dissertation, University Paris 3, 2004.