An Interactive Talking-Head

by

Dr Vincent E. Devin

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy

The University of Leeds
School of Computing
October 2002

The candidate confirms that the work submitted is his own and that appropriate credit has been given where reference has been made to the work of others.

Abstract

We demonstrate a novel method for producing a synthetic talking head. The method is based on earlier work in which the behaviour of a synthetic individual is generated by reference to a probabilistic model of interactive behaviour within the visual domain - such models are learnt automatically from typical interactions. We extend this work into a combined visual and auditory domain and employ a state-of-the-art facial appearance model. The result is a real-time synthetic talking head that responds appropriately and with correct timing to simple forms of greeting with variations in facial expression and intonation.

Acknowledgments

I would like to thank my supervisor, David Hogg, for his guidance and advice throughout this research.

I would also like to thank my colleagues of the Leeds Vision Group (past and present) for the stimulating working environment. Particular thanks to Afro, Chris, Derek and Hannah for checking my spelling and to Alison and Dan for their facial participation.

Thanks to Denis McKeown for his help to prepare the evaluation test and thanks to all the people who spent time doing it.

Declarations

Some parts of the work presented in this thesis have been published in the following articles:

Vincent E. Devin and David C. Hogg. "Reactive Memories: An Interactive Talking-Head". In *Proceedings British Machine Vision Conference*, volume 2, pages 603–612, BMVA, September 2001.

Vincent E. Devin and David C. Hogg. "Applications for a Generative Model of Joint Behaviour". In *Proceedings of CUES 2001, CVPR workshops*, BMVA, December 2001.

Contents

List of Figures

Chapter 1

Introduction

1.1 Interaction with the machine

A personal computer is now an object present in everybody's home. We use it as often as our televisions and our refrigerators. We work with it, we play with it but still consider it as a cold tool.

Recent progress in graphical display, audio production and computational speed allow computers to reproduce 'reality' with very high quality. Nevertheless, the relationship we have with them is still really basic. The interfaces we use haven't changed a lot for the past 20 years and we are still very far from a real human like interaction (Figure 1.1).

A 'talking head' is the most human display a computer could have to encourage the user to interact with it. Not only it has to look and sound real, it also has to have a human sense of behaviour based on facial expression, voice intonation and timing.

1.2 Motivations for an Interactive Talking Head

Proper human behaviour is something that can not easily be taught or described with rules. People learn how to interact by watching and participating. We believe it is the

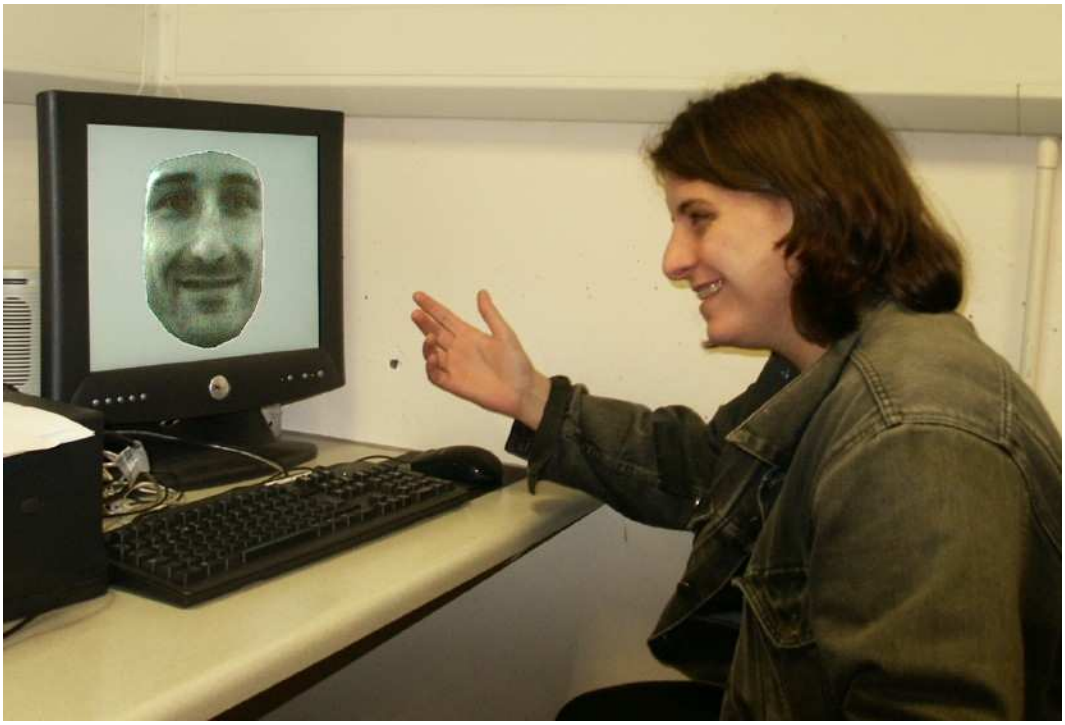


Figure 1.1: Is this how we interact with our computers?

same for the machine. Of course the content of what is said is important. However, the sequence and timing of accompanying facial expressions is also important; mistimed or inappropriate expressions may convey unintended meaning and can therefore be disruptive. It is reasonable to suppose the same requirements will apply for human interaction with a synthetic talking head.

The screen-based talking head is a powerful device for mediating interaction between humans and machines, enabling a form of interaction that mimics direct communication between humans [65, 8, 75]. The experience of realism is further enhanced when the computer is equipped with visual and auditory senses with which to perceive the user and react appropriately [48, 13, 97, 37]. In this symmetric situation, both the human and synthetic head can see and be seen, and can hear and be heard.

Usually based on rules manually set [101, 100, 26, 79, 82, 13], most of the talking heads look highly synthetic [78]. Some attempts have been made to automatically learn through observation interactive partners that have appropriately timed responses [48]. These char-

acters however were not as evolved as a talking head. This is exactly where the core of this research lies.

This thesis adapts a learning approach to interaction proposed by Johnson *et al.* [51] and by Jebara *et al.* [48], and extends this to deal with spoken as well as visual modalities within the same unified framework. Four applications of such a model are proposed and experimental results are presented.

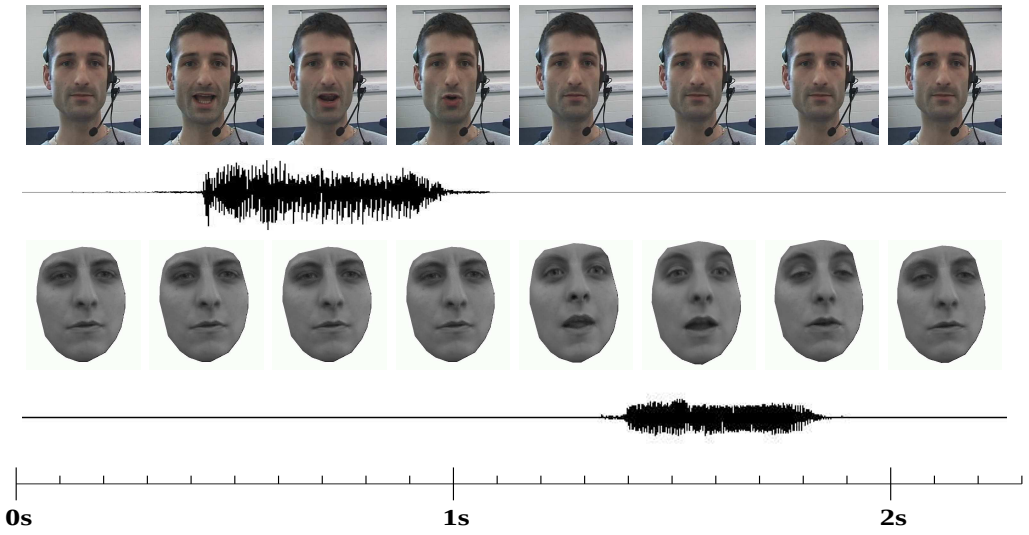
Figure 1.2 illustrates some results obtained with the interactive talking head. The Interactive head responds appropriately and with natural timing to each new input (tracked face and sound). More experiments are shown in Chapter 5.

1.3 Approach

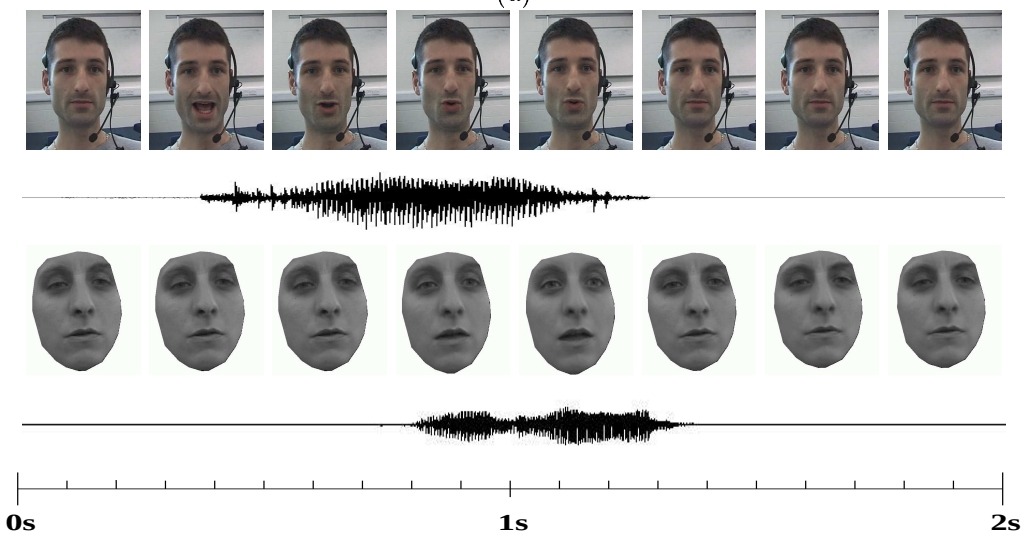
The interactive model is learnt from observation of human conversation. Speech and facial expressions are acquired and compressed into an internal representation from which this can be regenerated.

From a database of observation (the training set), the learning system creates a model that can be used to generate new behaviours (see Figure 1.3).

An approach that begins to meet these requirements is proposed in [51] and [48]. Their idea is based on the common notion of a state space, in which each vector represents the instantaneous configuration of a participant in an interaction. Such vectors are the end-point of a perceptual process within the computer, sensing the human party, and the start-point for a graphical process generating the synthetic individual. An interaction can be thought of as a pathway through the joint configuration space corresponding to the human and synthetic party in an interaction. The range of possible interactions is represented as a stochastic process over the joint configuration space, which is learnt through observation of real interactions captured on video. Johnson *et al.* [51] modeled the profiles of two people shaking hands. Jebara *et al.* [48] modeled head and hand gestures. In both cases, the models were used to drive a synthetic individual in response to past joint behaviour.



(a)



(b)

Figure 1.2: In (a) and (b) : On the first row, the face as input saying "hello" with the associated speech waveform, and on the second row, the synthesised face with the associated speech waveform.

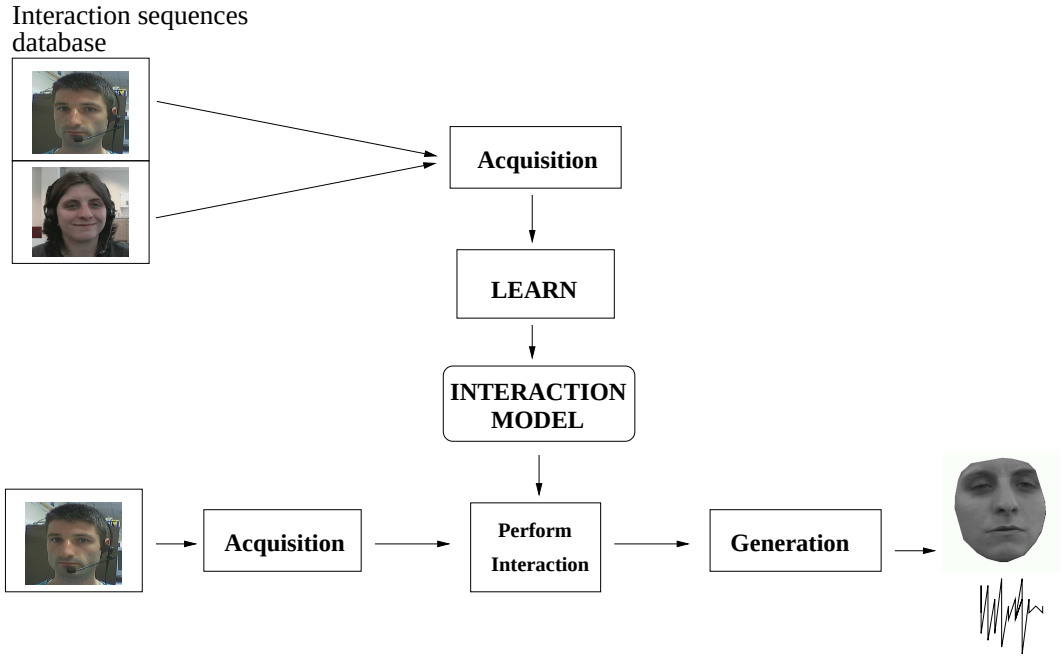


Figure 1.3: Learning by observation

In our case we deal with facial appearance and speech to create a talking head behaviour model.

1.4 Outline of the thesis

This introduction presented our approach on how we should interact with computers and the need for a new kind of interfaces.

Chapter 2 describes techniques that are related to the modelling and use of 'talking heads' as interfaces between humans and computers. It also describes our original research undertaken including descriptions of related background techniques. The remainder of the chapter is a description of current state of the art research in component areas such as face tracking and speech processing.

Chapter 3 presents the specific techniques used for the video and sound acquisition. The face tracker and face generator is presented as well as the speech analyser and generator.

Chapter 4 describes behaviour modelling. This is the heart of the interactive talking head application. The model is built from a training set of human interactions.

The use of a learnt behaviour model to generate a talking head is presented in Chapter 5. This chapter introduces four applications using the behaviour model with experimental results. It also includes an evaluation of the talking head results.

Finally, in Chapter 6, the thesis is summarized. The results are discussed and some pointers for future research are given.

Chapter 2

Background

Due to increases in processing power, especially in the domain of image and audio processing, the dream of being able to converse with a machine as if it were a human is now theoretically possible. The idea of a virtual interactive partner - a talking head - has been the subject of many projects. In the first section of this chapter we propose an overview of what has been done in the **talking heads** field until today, including the Leeds interactive behaviour system which is the base for the interactive talking head described here. To be able to converse with the machine we need to track voices and faces and process them. The following two sections present a short overview of the **Face Tracking** and **Speech processing** field.

2.1 Talking Heads

A study by Takeuchi and Nagao (of the Sony corporation) [97] demonstrated that users feel more comfortable interacting with a human like partner on screen than manipulating windows and reading text on a screen. Often, the aim of a computer interface is to display as much information as possible in the simplest way possible and to improve the communication between the user and the machine.

Human communication is composed of different methods: words, voice intonation and

Abilities			
Type	Interaction	Behaviour rules	Learning
I	X	X	X
II	√	X	X
III	√	√	X
IV	√	√	√

Table 2.1: Interaction abilities of each type of Talking Head

body language, see for example [73]. In a normal human conversation, if one try to deliver a message, it has been shown [73] that body language is 55% effective, intonation 38% effective and words 7% effective. What you say is not nearly as important as how you say it. So in H.M.I. (Human Machine Interaction), to communicate the best with the user, the use of a visual talking head is an obvious choice.

Solutions to the problem can be classified into four types, with increasing sophistication based on the ability to interact, to think and to learn (see Table 2.1). We shall call these type I, II, III and IV talking heads.

- Type I : a **Non-interactive talking head**. An artificial character (realistic or not) which talks and shows some visual information (Figure 2.1).
- Type II: an **Interactive talking head**. An artificial character can interact with the user via sensors. Even if the interaction is very basic the user can communicate with the machine and get a real time reply adapted to the situation (Figure 2.6).
- Type III: a **Rule-based interactive talking head**. A proper human conversation is possible. The machine is able to interact like a thinking entity by utilizing a set of artificial behaviour rules that are manually set (Figure 2.7).
- Type IV : a **Learning interactive human**. Much of our interactive behaviour is unconscious, therefore difficult to manually encode as 'behaviour rules'. The machine should learn, by observation, how to react like a human being and create those behaviour patterns we don't even realise we're obeying ourselves (Figure 2.9).

2.1.1 Type I : Non-interactive talking head

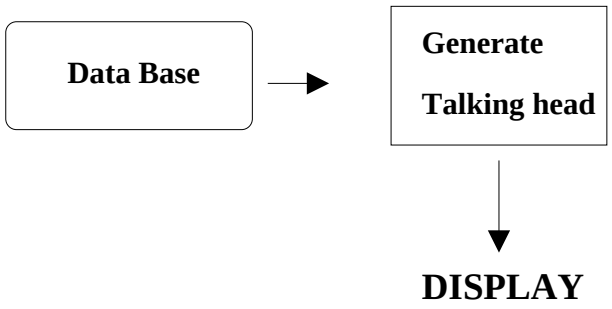


Figure 2.1: Non interactive synthetic talking head framework

Many synthetic talking heads are created every day for different purposes (broadcasting, business meetings, chat channels, video games). Many laboratories are actively attempting to develop better ways to create them. A survey on facial modeling and animation techniques by Jun-yong Noh and Ulrich Neumann [69] describes much of this work. These type I synthetic talking heads do not interact but try to communicate information as realistic as possible with a displayed head.

One of the first talking heads producing speech from text and audio was created by Morishima [65]. The mapping from speech to audio is achieved in two different ways. One is a rule-based phoneme to facial movement system, manually set, mapping input phonemes coded from either audio speech or text. The other, from pre-learnt codebooks encoding LPC (Linear Predictive Coding) and corresponding facial position selected via a vector quantisation of the linear coefficient of the new audio input.

A plausible approach has been to animate synthetic characters from real human facial expression [71, 70]. The technique is called 'expression cloning' (see Figure 2.2). The resulting synthetic talking head is silent as the speech has not been included yet. The 3D synthetic model is animated from feature points selected on real subjects. The best mapping between the real feature points and the synthetic ones is represented by radial basis functions.

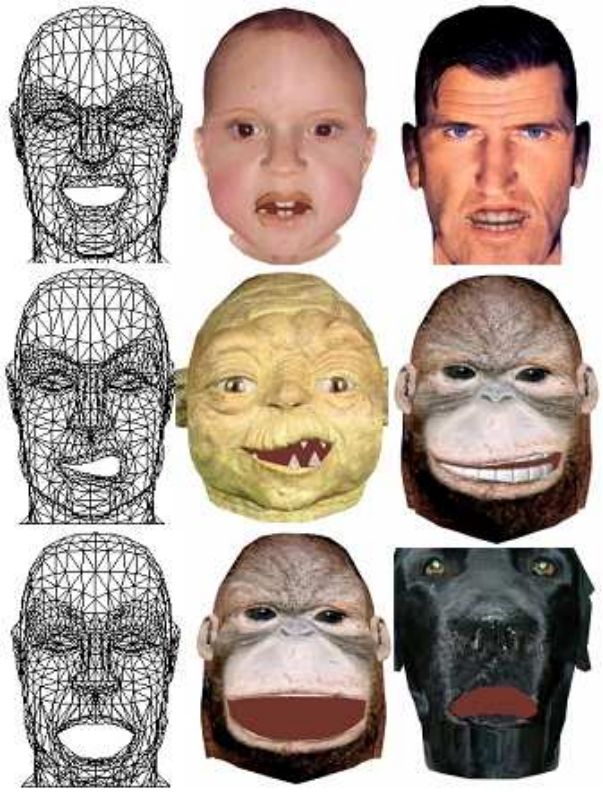


Figure 2.2: Expression cloning

Cosatto and Graf are working on a photo realistic head created from video footage of talking faces [20, 34]. 2D features of real faces are extracted to create the model. The synthetic head is driven by a real talking person whose characteristic feature points are tracked with correlation methods. Audio is integrated at the end with a text to speech synthesiser.

A photo realistic 3D head from photographs is proposed by Pighin [80]. The 3D face models are created from 2D pictures of real people. To generate the 3D model they adapt a generic 3D face to a specific individual face photograph. Feature points are detected and matched on each 3D model. The different facial expressions are achieved by morphing between those models (see Figure 2.3).

Olives *et al.* created a synthetic talking head with a method of auditory speech synthesis [75]. A 3D head model, with characteristic points controlling the facial expression and lips

Figure 2.3: Synthesizing realistic facial expressions from photographs

movement is manually set to match each of the phonemes used in Finnish. Those matches are called 'visemes'. The talking head is rendered by interpolating between the visemes.

Brooke has worked on talking heads for a better understanding of speech in computer interfaces [8, 9]. The model learns, with HMMs (Hidden Markov Models), low definition grey level talking faces compressed with PCA (Principal Component Analysis). The talking faces are regenerated using the learnt HMM.

Bowden *et al.* [6] create a humanoid face, Jeremiah, looking at moving objects in a movie. The object detection is based on a background subtraction. To model the variation of the background the system builds a Gaussian mixture of the colour distribution for each pixel. From simple parameters extracted from moving objects the system sets Jeremiah's emotion to preset facial expressions such as boredom, surprise or happiness, see Figure 2.4. The 3D head is an eigen-model built from motion keyframes. The eyes looking at the moving objects are controlled separately.

The most famous talking head on the Web is probably ANANOVA, the world's first virtual newscaster (see Figure 2.5) who delivers the news every day on the net (and soon on mobile phones). The animated character uses a text to speech converter with some tags manually inserted for the intonation and emotion it is supposed to show. The animation model then fits the speech with the corresponding facial movement which are manually set for each phonemes of the english language.

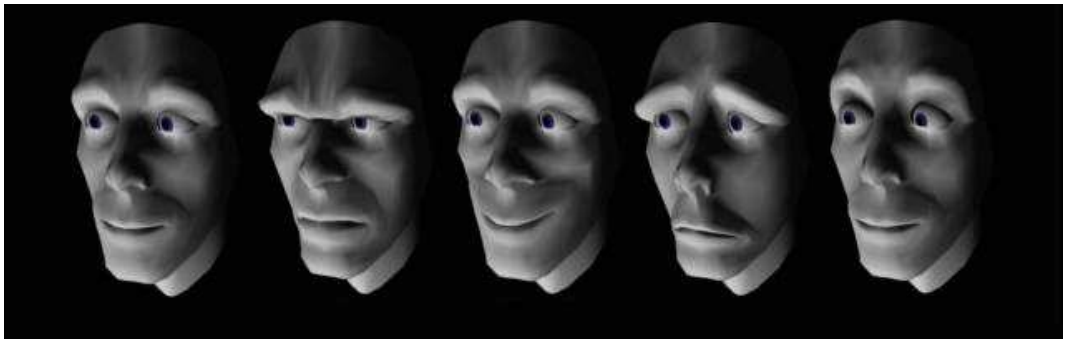


Figure 2.4: Jeremiah's facial expressions

Figure 2.5: Ananova (www.ananova.com) and one of her followers, Chase Walker (Sprint advanced technology lab).

2.1.2 Type II : Interactive Heads

This second type of interactive head allows feedback from the displayed head to the user. This loop produces the talking head's reply to the user input (see Figure 2.6).

Takeuchi and Nagao of the Sony corporation performed a study to understand the differences between interacting with a machine with or without facial displays [97]. Their conclusion was that the new modality helps the conversation to go forward and makes the user more talkative. This modality needs a feedback from the synthetic character to the user. The project was to create interactive booths to sell Sony's products using a talking head.

Hasegawa created an interactive head which responded to finger gestures and facial identity

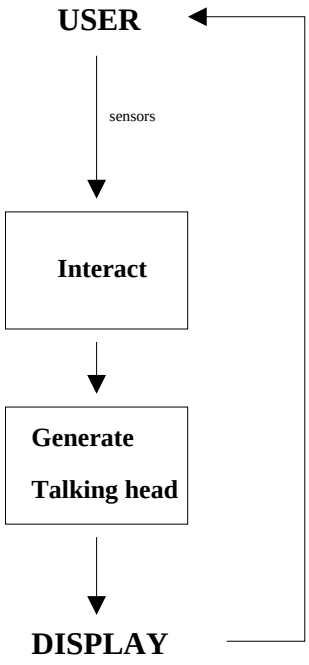


Figure 2.6: Interactive Head framework

[37, 38]. The displayed head recognises the palm sign and finger motion and gazes at the hand with changing facial expressions. The tracking method is based on skin colour detection and the face recognition uses template matching. The size of the tracked zone gives the information to discriminate a finger gesture or a palm sign. The displayed head is generated from mapped polygons. Parameters controlling the eye gaze, the eyelids and lips, move along with the results of tracking.

2.1.3 Type III : Smart interactive talking head

This third type of talking head allows a more complex interaction with the user. The interactive talking head obeys a manually set model and chooses the way it interacts with the user. By using a set of behaviour rules the interactive partner can be human like and a conversation is then possible.

Thorisson, from the M.I.T Media lab, created the interactive character 'Gandalf' which converses with the user using speech, facial expressions and hand gestures (see Figure 2.8).

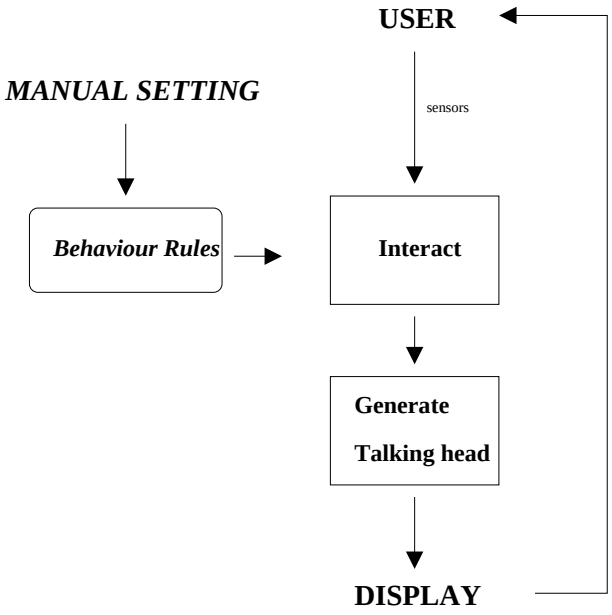


Figure 2.7: Smart interactive talking head framework

The user is equipped with gloves, helmet and various other kinds of sensor. Hardware pitch recognition and software word recognition are used for the speech recognition. Gandalf's behaviour rules for human face-to-face conduct come directly from the psychological literature on human-human interaction [101, 100].

Figure 2.8: Gandalf, the interactive cartoon, from the M.I.T. Media lab.

Pelachaud, working with Poggi, De Carolis, Cassel, Pasquariello and Cappella have undertaken extensive research on the use of behaviour rules in an interactive partner. The aim was to create a talking head able to send complex 'natural' messages and emotional meaning to users based on cognitive rules [78, 26, 79, 82, 13]. Everything is set manually and the cognitive rules control every feature of the face and voice intonation of the character. The latest talking head resulting from these researches is 'Greta' [78] a 3D synthetic head.

2.1.4 Type IV : Learning interactive human

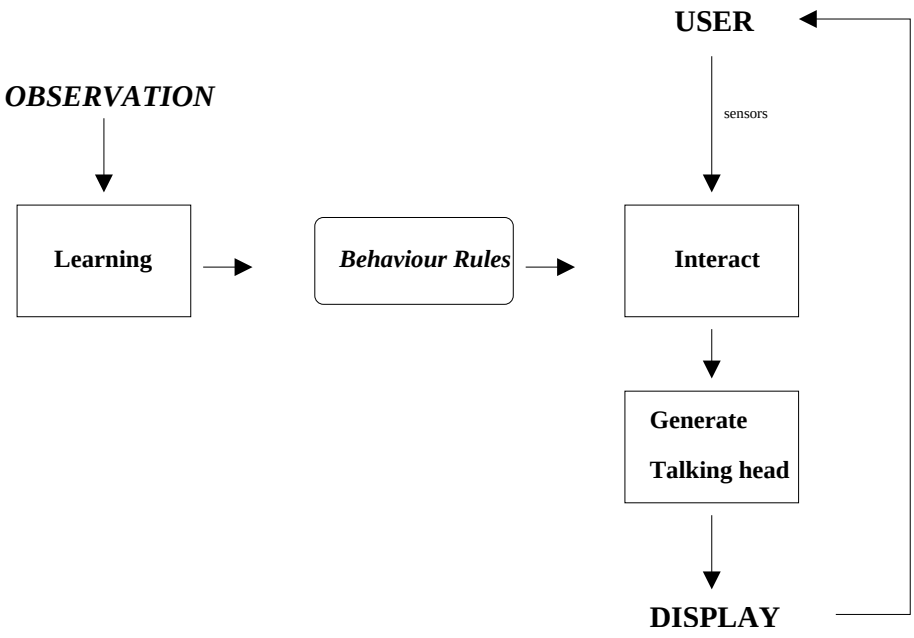


Figure 2.9: Learning interactive human framework

Much of our interactive behaviour is not apparent even to ourselves. To create this unconscious behaviour we, humans and machines, need to observe people interacting and grab those little things we do which make the conversation easier and richer. This fourth type of interactive talking head learns, like humans, from observation.

Tony Jebarra proposed an approach for analyzing and synthesizing human behaviour using a time series of perceptual measurements. The result is a gesticulating interactive partner

Figure 2.10: The interactive learning partner

(see Figure 2.10) which learns how to respond to a user [48]. This model automatically learn from long human interactions. The interaction is produced by an 'action-reaction' learning system which deals with multi-dimensional time series. The multi-dimensional series is a concatenation of the successive values of parameters describing the head and hands, obtained from tracking for each interacting individual as a set of coloured blobs. The size of the vectors are 30 parameters (6 Gaussian blobs).

The preprocessing rasterizes the times series into vectors (see equation 2.1) by sliding a time window covering a fixed interval (typically 6.5 seconds, given 128 frames at 50ms intervals in the example shown) .

$$V = \{C_1, C_2 ... C_T\} \tag{2.1}$$

where

$$\begin{aligned} C_i &= \{H_i^l, H_i^r\} \\ H^l &= \text{Head and Hands blob tracking results of the left individual} \\ H^r &= \text{Head and Hands blob tracking of the right individual} \\ T &= 128 \text{ (6.5 seconds)} \end{aligned}$$

PCA is then performed on the resulting vectors to obtain the most energetic modes of variation (40 modes were found to give more than 95% of the variance). Each period

of time is then generated with a reduced number of parameters and concatenated to the most recent vectors in the time series. Those vectors can now be associated with the next event.

Time series of configuration vectors given by $C_1, C_2 ... C_t$.

Now encode T samples : $C_{t-T} ... C_t$ to give :

$$A_t = \{a_1 ... a_4 0\}_t \tag{2.2}$$

Now form pairing :

$$\{(A_t, C_t), C_{t+1}\} \tag{2.3}$$

With those pairs of Action (A_t, C_t) Reaction (C_{t+1}) they can learn using Conditional Expectation Maximization (C.E.M. [47]) how to predict and generate a virtual reaction for a virtual partner.

The creation of a realistic talking head which creates its own set of behaviour rules via automatic learning methods is a project built from different researches at the University of Leeds. This talking head is a type IV learning interactive human. It automatically learns from visual and audio observations. This application is built from previous work.

Johnson and Hogg's work on learning the distribution of object trajectories [52] (see Figure 2.11) proposed a way to describe temporal activities. Their approach is related to Bobick and Davis model of a temporal template for recognition of human movement [24].

The idea is to represent the conditional probability density function $p(C_t|C_{t-1})$ where C_t is a Configuration vector at time t.

The method is in two parts. First the state vectors derived from the behaviour training data are approximated by a set of state prototypes placed by vector quantisation (see equation 2.4).

Figure 2.11: From observation of many trajectories the system is able to predict the path of a pedestrian.

$$\{S_1, S_2, ..., S_v\} \tag{2.4}$$

where

$$\begin{aligned} S_i &\in VQ(C_1, C_2 ... C_k) \\ C_i &= \{H_i^l, H_i^r\} \\ H^l &= \text{Tracked silhouette of the left individual} \\ H^r &= \text{Tracked silhouette of the right individual} \\ v &= \text{length} \end{aligned}$$

The second part builds on the first to provide a vector representation for extended behaviours. The way in which this is achieved may be visualized as follows. Each state prototype has an associated activation level that is initialised to zero for the encoding of a single behaviour. The behaviour is traced from beginning to end, and at each time step the activation of each state prototype either decays by a fixed proportion or takes on a value that is a linearly decreasing function of the prototype's distance from the current state, whichever gives the largest value. The pattern of activation levels at each time-step provides an encoding of the behaviour up until that point on the trajectory and this is recorded as a vector - referred to as a behaviour vector to distinguish from the state vectors.

Figure 2.12: Example of a suspicious trajectory in a car park

The same procedure is repeated for all behaviours in the training set, giving a large collection of behaviour vectors encoding these trajectories and all partial trajectories implicit in their generation. This approach to encoding the evolution of a system is equivalent to the so-called leaky neural network model [85, 107]. Finally, the distribution of behaviour vectors extracted from the training data are themselves encoded by a set of prototype behaviours, again derived by vector quantisation (see equation 2.5).

$$\{X_1, X_2, ..., X_m\} \tag{2.5}$$

where

$$\begin{aligned} X_i &\in VQ(V_1, V_2 ... V_k) \\ V_i &= \text{joint 'activations'} \end{aligned}$$

The final behaviour prototypes provide a compressed model for the range of behaviours observed in the training data. This model can be adapted to serve as a piecewise uniform probability density function in which each prototype is replaced by a uniform region with magnitude proportional to the local density of prototypes, which is in turn proportional to the observed density of training behaviours. A Markov chain is then superimposed on top of the behaviour prototypes, with transitions defining the ways in which behaviours in the neighbourhood of prototypes may evolve between time-steps. The probability of

a transition is estimated from the proportion of such transitions observed in the training set.

Figure 2.13: Leeds' first interactive partner

This work led to the first approach of an interactive partner [51] (see Figure 2.13). The model is learnt through observation of real interactions captured on video, described in Johnson [50].

Subsequently, different works on behaviour were undertaken :

- A surveillance application to detect abnormal behaviour in a car park [66], see Figure 2.12. The work was based on the prediction model of walking people.

- An extension of the model with grammar based on Variable Length Markov Models was proposed by Galata to model people doing aerobic exercises [32].

- The first interactive head which learns automatically how to react . First with basic silent expressions in Hogg *et al.* [41], then with speech in this research (see Figure 2.14).

2.2 Face tracking

The domain of face tracking is very active. Major surveys of this domain include Samal & Yiangar in Pattern Recognition [92] and Rama Chellappa [14], with more recent surveys by Hjelmas in 2001 [58] and Ming-Hsuan Yang, David Kriegman and Narendra Ahuja in 2002 [108].

Figure 2.14: Leeds Interactive Talking Head

In this section we produce a reduced and basic overview of the most commonly used methods in face tracking. A more accurate and complete description can be found in the surveys mentioned above.

There are two kinds of approach to face tracking : **Feature-Based** and **Image-Based**. The **Feature-Based** approach uses facial knowledge by locating and tracking specific prominent features. The **Image-Based** approach uses the face as a whole and can classify results into face and non-face classes. I will finish by describing the tracker I'm using in my application which can track heads and facial expression but also reproduce what has been tracked.

2.2.1 Feature-Based approaches

In the feature-based approach, low level features of the face are selected and used for detection and tracking. **Contour** tracking uses knowledge of the shape of the face or any other specific facial feature. **Intensity** techniques use the intensity value of areas of pixels composing the face. Motion tracking techniques locate moving objects by using displacement measures. **Feature** tracking uses prior knowledge of one or several specific prominent facial features and anthropometric face geometry.

2.2.1.1 Contour

A popular way to track faces from their contours is with an active contour method. This method requires a previous edge detection. The Sobel operators, the Marr-Hildreth edge operator or different first and second derivatives Laplacian of Gaussians [11] are commonly used. More recent edge-based techniques can be found in [25, 59].

The principle of an active contour is to deform a contour to take the shape of the face or feature to track. The first active contour method (called a 'snake') was introduced by Kass *et al.* [53]. The snake tries to minimize an energy function :

$$E_{snake} = E_{internal} + E_{external} \tag{2.6}$$

where $E_{internal}$ depends upon a predefined shape applied to the snake and $E_{external}$ describes the effect of the image after being processed (e.g., edge detection). The predefined shape depends upon the nature of the feature to track. By minimizing its energy the snake will stick to the contour and track the feature.

Terzopoulos improved this technique by adding a Kalman filter in the tracking [99], and later on Isard and Blake added the CONDENSATION method [46], an application in vision of the *particle filter* [33].

The PDM (Point Distribution Model) was created by Taylor and Cootes. This is also known as the 'smart' snake [18]

The PDM is a statistically compact description of the shape. The contour is discretized into a set of labeled points and extracted from a large set of faces. PCA is then performed to extract the principal modes of variation over all the contours:

$$x = \hat{x} + P_s b_s \tag{2.7}$$

Where $\hat{x}$ is the mean shape, P_s is a set of orthogonal modes of variation and b_s is a set of shape parameters.

The manual contour point setting was simplified by Hill in an automatic landmark technique [39]. The PDM approach may also be combined with the use of intensity information. This is know as the Active Appearance Model (described in Section 2.2.3)

2.2.1.2 Intensity

The colour information within a face can be used to locate and track it. A complete review of intensity tracking methods can be found in [58]. The most common colour model used to describe the skin is the normalized RGB representation as the skin appearance changes with the lightning conditions [7, 88, 68, 89]. The HSI (Hue, Saturation, Intensity) has been proven actually better than RGB by Hwan Lee Choong [15] and is now commonly used for face segmentation [34, 93, 63, 103].

A recent review has been undertaken by Terrillon in his comparative study of widely used colour spaces for face detection [98]

The location of the face can be done with a skin colour threshold where the skin colour is modeled through histograms [54, 55, 38, 34, 88, 68, 89]. The use of Gaussian distribution to detect skin colour is also used to get a more general spectrum of colour [74, 64, 22, 63]. The Gaussian parameters can be set up with colour samples taken from different ethnic groups.

2.2.1.3 Motion

In the case of sequences of faces the use of motion detection is an effective method to locate faces.

The detection of natural eye blinking [59, 22] is a nice way to locate the face using frame differences. Most motion tracking methods try to locate what is actually moving as a pre-tracking process. It is usually followed by a more sophisticated method such as pattern recognition or PCA performed on the new region of interest set by the motion detector [7, 63, 38, 88, 109, 103, 22]

A spatio-temporal Gaussian filter has been used by McKenna *et al* [62] to detect faces. The process involves convolution of a grey image $I(x, y)$ with the second order temporal edge operator $m(x, y, t)$ which is defined from the Gaussian filter $G(x, y, t)$:

$$G(x, y, t) = u \left(\frac{a}{\pi}\right)^{\frac{3}{2}} e^{-a(x^2+y^2+u^2t^2)} \tag{2.8}$$
$$m(x, y, t) = -\left(\nabla^2 + \frac{1}{u^2}\frac{\partial^2}{\partial t^2}\right) G(x, y, t)$$

The optical flow method is based on short-range moving pattern and is sensitive to fine motion. The method is modeled by the image flow equation :

$$I_x V_x + I_y V_y + I_t = 0 \tag{2.9}$$

Where I_x, I_y and I_t are the spatio-temporal derivatives of the image intensity and V_x and V_y are the image velocities. By solving this equation for V_x and V_y an optical flow field that contains moving pixel trajectories is obtained. Then a classification of moving and non-moving region has to be done from those trajectories. Lee *et al.* proposed a line clustering algorithm [15]. This method is a faster version of the original algorithm by Schunck [87]. Optical flow is now a widely used algorithm [25, 27, 34, 88]

2.2.1.4 Features

Locating specific features is a common and effective way to track a face. The method is used as it allows to create fast applications [86], but also because of its reliability. If used with multiple features this method is relatively insensitive to occlusions. People usually locate the eyes [22, 40, 89] but all the other prominent features of the face are valid [103, 93, 42, 10, 96, 27, 7, 4, 34, 88, 109]. Features are usually located via correlation methods. An experimental comparison of correlation techniques can be found in [23].

For a better location of these features, some anthropometric measures of the face are usually used [49, 25, 11, 10].

Feature location is a powerful tool for face tracking but also can be used for person recognition or even expression recognition. Usually the imaged-based approach (2.2.2) is more common for that kind of detection. Brunelli discusses and experiments those methods in his comparison between features and templates [11]

2.2.2 Image-based approaches

The **Image-Based approach** considers the task of face tracking as a pattern recognition problem. The face is taken as a whole and can be classified into face and non-face classes. Usually used for expression and person recognition it also has some very good tracking properties. While some people use basic template matching [57, 42] with the methods described in 2.2.1.4, the face description is often made in a linear subspace.

Although many different approaches exist, PCA is by far the most common method used, involving linear subspaces.

Used in the 1980s by Sirovich and Kirby [91] to represent faces it has been developed later by Turk and Pentland [104] for face recognition. Given a set of training images an optimal subspace is determined such that the mean square error between the projection of the training set onto this subspace and the original images is minimised.

For example we have a dataset of n faces, $F_1, F_2, ..., F_n$. The average face is defined by

$$\Psi = \frac{1}{n}\sum_{i=1}^{n} F_i \tag{2.10}$$

$$\Phi_i = (F_i - \Psi)^{\nu} \tag{2.11}$$

where Φ is the image vector for each image of the training set. Then with $D = [\Phi_1\Phi_2...\Phi_n]$ and $C = DD^t$ we compute the eigenvectors u_i of C. An input image can be projected onto face space by

$$\omega_k = u_k^t \Phi \tag{2.12}$$

where $k = 1...m$ and m is the number of principal components selected.

This method has been used in much research [63, 64, 1, 22, 88, 28, 19, 17, 36, 29, 27, 21, 3] for face tracking, encoding and compression.

It has been extended for face discrimination (Linear Discriminant Analysis) and factor analysis and into the Active Appearance Model described in the next section.

2.2.3 The Manchester Face Tracker

Cootes created a face tracker based on the Active Appearance Model [17]. The algorithm can locate faces and their expression, track them and reproduce the sequence from reduced size vectors obtained at the tracking.

First a PDM is created on a training set of moving faces. Then PCA is performed on the grey level images of the same set of moving faces (see figure 2.15).

Figure 2.15: Grey level images

The Active Appearance Model is a combination of both PDM and grey level PCA into one single model. The tracking of a new sequence of faces is done by deforming and matching the face appearance model to the input faces (see Figure 2.16). The process is explained

in [17]. Results and more details can be found in the next chapter, as this method is the one we use to create the visual aspect of our Talking Head.

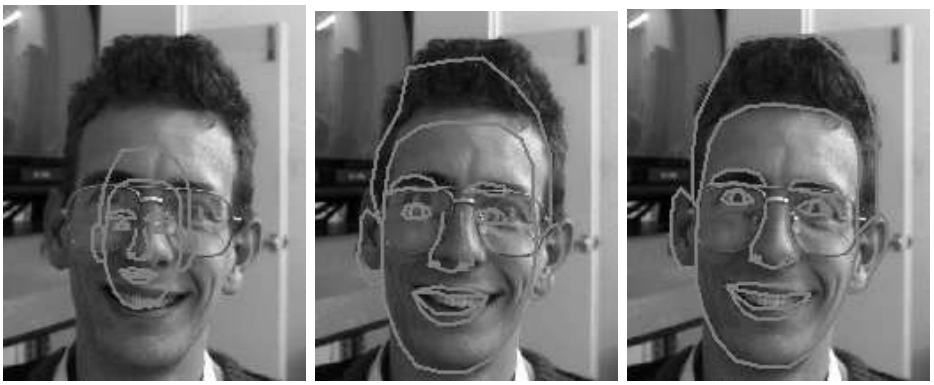


Figure 2.16: Face matching

Since the creation of the Active Appearance model extensions have been made. Recently Hou *et al.* proposed the Direct Appearance Model [43] in which an improvement of 3% in the tracking quality is obtained. Walker, from the Manchester lab presents another extension to the original method with an automatic way to build the AAM using salient features [106]

2.3 Speech Processing

Speech processing is another active research area. An overview of this field is presented in Cole [16]. Only a subset of this field is relevant to the work presented here, specifically the areas of *Coding and Compression*, *Analysis*, and *Generation of speech.*

2.3.1 Coding and Compression

Until recently, research into speech coding and compression was split in two: vocoders and waveform coding.

The first deals with knowledge of physical speech production and manipulation of many

parameters with great precision. The second one tries to reproduce the signal itself in the time or frequency domains. Each of these fields has a different goal. Vocoders achieve considerable bit rate savings at the cost of quality whilst waveform coders preserve quality but do not compress speech to any great extent. For a good overview of this early research, see [30].

More recently, the appearance of published work on speech coding and compression has increased a lot and can not be easily classified into those two fields.

In the frequency domain the Fourier transform is the most commonly used technique for any kind of speech preprocessing [90]. The selection of relevant frequencies is a start for a compression or simple encoding of the speech.

Another frequently used technique is that of linear predictive coding [61]. This forms the basis for many compression algorithms, such as the CELP (Code Excited Linear Prediction) [12] which adds a residual error compression to the normal encoding of the LPC algorithm [102].

This method can also be combined with other techniques, some of which are well known in the vision field, for example Vector Quantisation [60]. The LPC coefficient vectors can be quantized for a better compression of the speech and the resultant compression algorithm is known as lpc10 [76]

Another technique well known in the vision domain is principal component analysis. PCA is used to remove linear dependencies between sets of coefficients [45]. The use of PCA in speech processing will be described and evaluated in the next chapter.

2.3.2 Analysis

Speech Analysis is one of the biggest (if not the biggest) sub-parts of the Speech processing domain. It would be foolish to try to present an overview of the entire field, even a quick one, in a few paragraphs. For a complete one see [16].

The Hidden Markov Model (HMM) is probably the most commonly used method to de-

scribe and analyse speech [83]. The HMM is a combination of two stochastic processes. An observable one, which in the speech recognition domain accounts for spectral variability, and a hidden one, which in the speech recognition domain accounts for temporal variability. The structure of the observable Markov chain is manually set and has to be carefully chosen. The model is then trained on a dataset to set the parameters of the hidden process.

Speech analysis is typically based on one or more of the encoding methods described in the previous section. After being encoded and compressed the parameters of the speech description can then be easily compared and analysed via various statistical methods. LPC [61], Fourier Transform [90] and PCA [45] are all valid encoding methods for speech.

Another major issue within this field is that of pitch extraction. Most speech processing needs pitch intonation information to complete the analysis [81, 84]. Also, it is worth noting that working only with the voice intonation can be enough for some applications such as speaker recognition. The aim of a pitch extractor is to find the fundamental frequency (F_0). A description of that frequency and some of the most common ways to extract it can be found in [16, 94, 72].

Approaches to speaker adaptation are similar in principle, except that the models are more commonly general statistical models of feature variability, rather than models of the sources of speaker-to-speaker variability [44]. Whereas the speech analysis field is very important to the speech community this area does not concern us. All the processing we will do is part of the statistical model we are using. On the specific 'audio' aspect of our work, we will be going straight from encoding to generation.

2.3.3 Generation

The generation of speech is the final part of the three stage process we adopt in the 'audio'. Some books and overviews of the field can be found in [31, 105, 16].

The '*learn and reproduce*' approach is not really used here. Most of the work in this domain has a pre-analysis stage and the output is always what is understood by the system. Audio

output is usually generated from text.

Three different approaches to generating audio output can be found in the literature. These are *articulatory synthesizers*, *formant synthesizers* and *concatenative synthesizers*.

Articulatory synthesizers are physical models based on the detailed description of the physiology of speech production and on the physics of sound generation [77].

Formant synthesis is a descriptive acoustic-phonetic approach to synthesis [2, 95]. Speech generation is performed by modeling the main acoustic features of the speech signal. The basic acoustic model is the source/filter model. The filter, described by a small set of formants, represents articulation in speech and the source represents phonation. Both source and filter are controlled by a set of phonetic rules.

Concatenative synthesis is based on speech signal processing of natural speech databases [67]. The database includes the major phonological features of a language. The synthesizer concatenates speech segments, and performs some signal processing to smooth unit transitions and to match predefined prosodic schemes.

Our approach is close to the *concatenative synthesizers* as you will see in the next chapter.

An evaluation of the quality of generated text-to-speech has been carried out in [56]. Human subjects were asked to assign a quality mark to generated output and express their opinions via multiple choice questionnaires. We are using a similar method to evaluate the quality of our results, as reported in Chapter 5.

In this chapter, the need for an automatically learnt talking head (type IV) has been demonstrated. The remainder of this thesis will describe how this talking head is created.

Chapter 3

The acquisition of data

This chapter describes the methods used to acquire data for entry into the database of interaction sequences needed for the learning process. This learning process is described in Chapter 4. These acquisition methods are also used as direct input for the applications described in Chapter 5

To create the database of interaction sequences, the video and sound are processed separately before being stored. Both of these processes, the 'face tracking' and the 'speech analysis', use a model based compression to encode the signal into low-dimensional vectors. The aim is to be able to store them easily and reprocess them to regenerate the original signals. Figure 3.1 illustrates this process.

The data acquisition set comprises pairs of people talking to each other using cameras and microphones in a setup similar to that shown in Figure 3.2. All of the interaction is recorded on digital video tapes and processed. The filming is staged so the speakers are greeting each other or having a conversation, depending upon the application and the kind of sequences desired (see Chapter 5).

The video is processed by the **face tracker**, described in the first section. The sound is processed by the **speech analysis**, described in the second section. The result of both face tracker and speech analysis will be combined into one single vector to be processed by the behaviour modelling system, described in the next chapter.

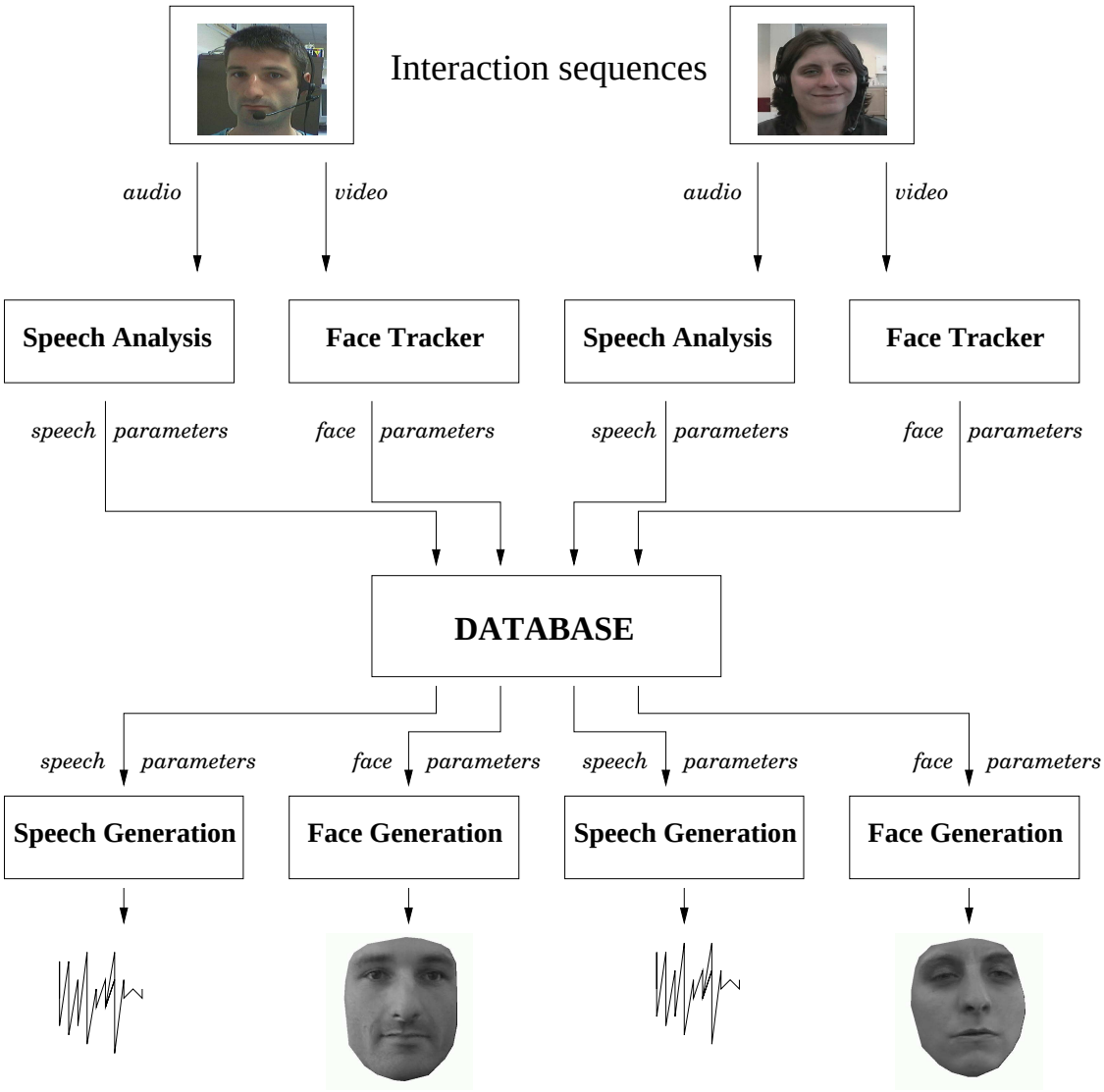


Figure 3.1: Data acquisition and generation process

3.1 The Face tracker

To encode the face into a low-dimensional vector, an existing method based on a deformable appearance model [17] is used (see Section 2.2.3). The purpose of the tracker is to extract, using a preprocessed face model, the facial parameters. These parameters can be stored and reused to recreate the face tracked (see Figure 3.3).

Facial appearance is represented by the parameters of a combined model of shape and

Figure 3.2: The experimental setup for video capture of interaction sequences

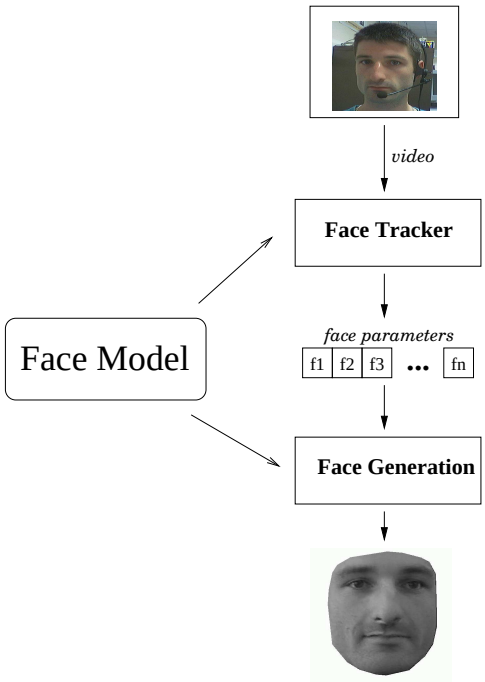


Figure 3.3: Encoding the face

intensity variation for the human face. This model is generated from training data of typical faces, marked-up by hand with spline curves delineating prominent facial structures

(Figure 3.4). Applying principal component analysis (PCA) to the set of parameters describing facial lines within an aligned frame of reference, recovers an underlying set of axes of shape variation.

Any face shape x can then be approximated with :

$$x = \bar{x} + P_s b_s \tag{3.1}$$

where $\bar{x}$ is the mean shape, P_s is a matrix with columns which are the principal orthogonal modes of variation and b_s is a vector of shape parameters. Figure 3.5 illustrates the facial lines generated from three different sets of values for the parameters of such a shape model.

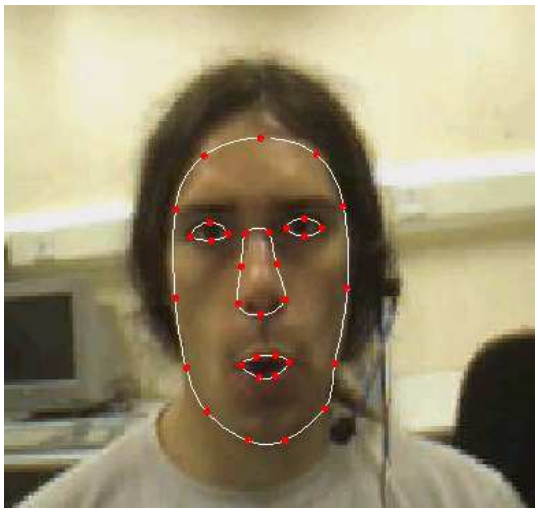


Figure 3.4: Spline curves delineating prominent structures of the face

A grey-level appearance model is constructed by warping each face from the training set onto the mean shape and applying principal component analysis to the normalised data. The warping is performed by triangulating between points on each facial line, as shown in Figure 3.6, and applying an affine mapping between corresponding triangles. Any normalised grey-level face g can then be approximated with :

$$g = \bar{g} + P_g b_g \tag{3.2}$$

where $\bar{g}$ is the mean grey-level face, P_g a matrix with columns which are the principal orthogonal modes of variation and b_g a vector of grey-level parameters.

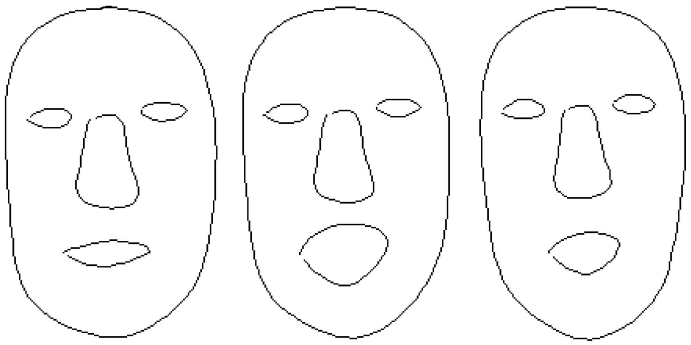


Figure 3.5: Face shapes generated from three different sets of parameter values

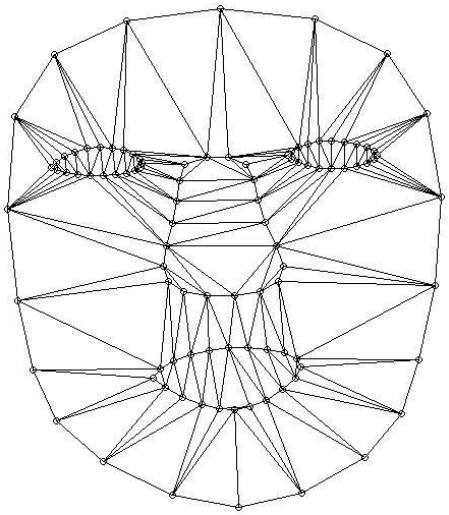


Figure 3.6: Mean shape triangulated for warping

A face can now be synthesised from shape and grey-level parameters by generating the normalised grey-level image and warping it to the given shape (Figure 3.7).

Different models are used for different individuals and even different applications. In the experiments which are reported in Chapter 5, there are between 4 and 10 parameters in the final facial appearance model to which 4 parameters, X, Y, S and R for position, scaling and rotation are added. Some variation of the affine parameters can be seen in Figure 3.8.

A separate model is required for each individual. Variations in identity are not modelled, although the modeling framework can be extended to do this [17]. For better results,

Figure 3.7: Faces generated from three sets of values for model parameters

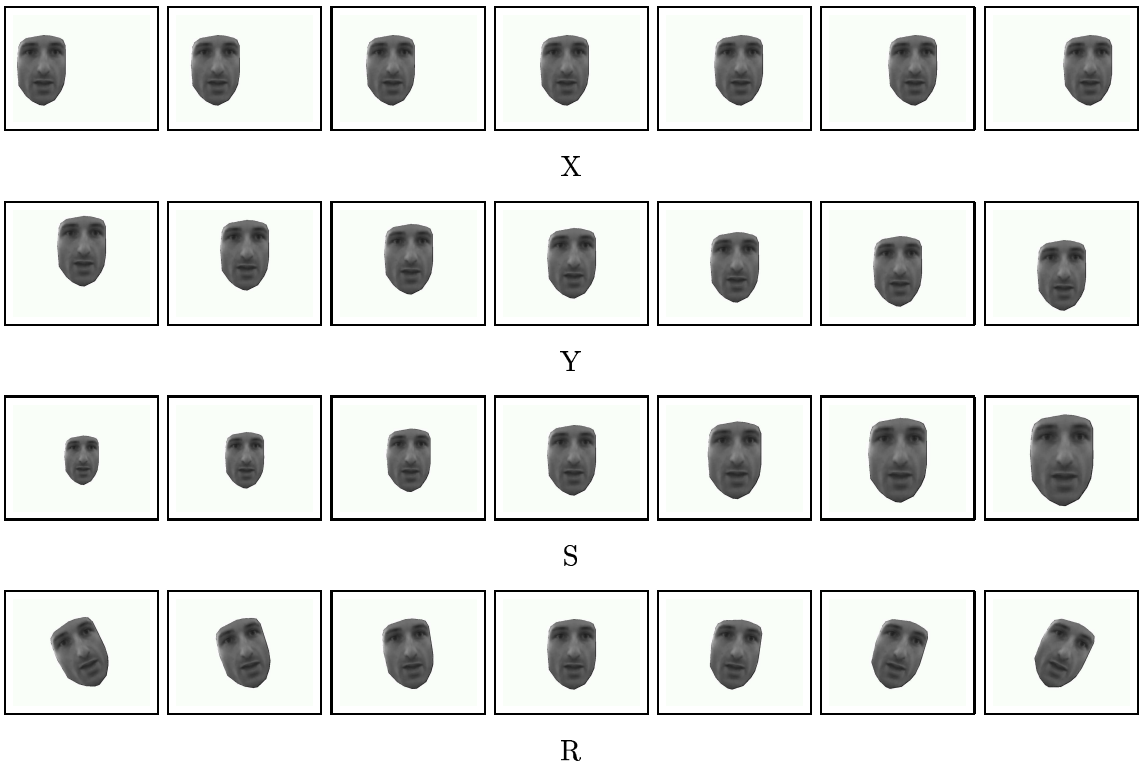


Figure 3.8: Varying the four affine parameters. Position X in first line, position Y in second line, scale in third line and rotation R in fourth line

the face model is created to render a limited number of expression. Results are shown in Chapter 5. Note that to model the same kind of expression with different individuals the number of parameters needed can vary due to more or less complex facial features. Figures 3.9 and 3.10 illustrate some individual characteristics which had to be included in

the model. The speaker in Figure 3.9 has very animated eyes, her eye movements convey much information. The movement of the eyelids of the speaker in Figure 3.10, is part of her way to answer even very basic greetings. Both of these special kinds of facial expressions have to be included in the face models.

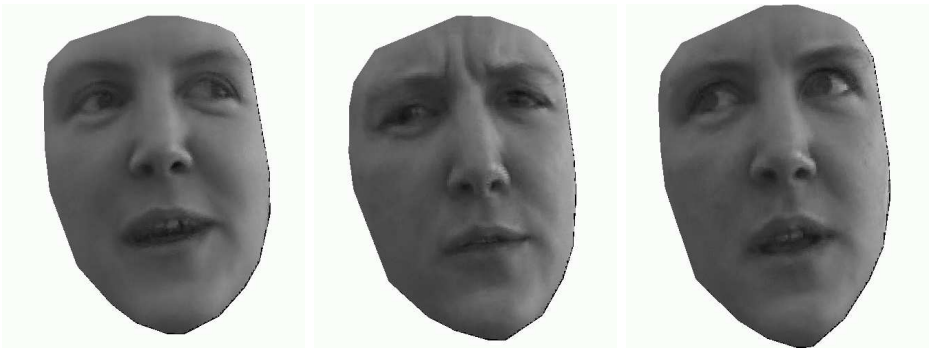


Figure 3.9: The eye movements of the speaker are encoded in the model

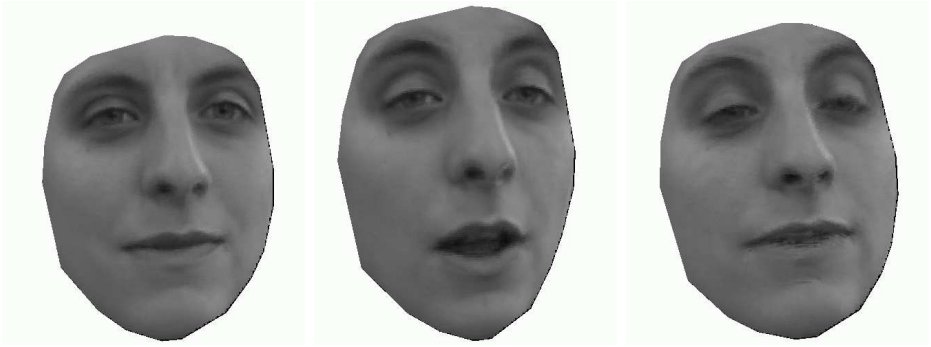


Figure 3.10: The eyelid movements of the speaker are encoded in the model

To avoid any correlations between shape and intensity variations, shape and grey-level parameters are joined together and another PCA is performed. A specific face is represented by assigning values to the n parameters of the facial appearance model: $\mathbf{f} = \{f_1, f_2, ..., f_n\}$.

The mapping from a given visual image to the corresponding parameters of the appearance model is performed using the iterative search proposed by Cootes[17]. The idea is to adjust model parameters so that the corresponding synthetic face matches the new image as closely as possible. Figure 3.11 shows a close match between an input image and the

synthetic face chosen as best match.

(a) (b) (c)

Figure 3.11: The closest matching synthetic face to a new image. (a) input frame, (b) best matching, (c) contour of best matching superimposed on input image.

To find the best match, the difference between a new image and one synthesised by the facial appearance model needs to be minimised. Figure 3.12 shows the difference image obtained by subtracting the synthesised image from the new video image. A difference vector δI can be defined:

$$\delta I = I - I_s \tag{3.3}$$

Figure 3.12: Subtracting the synthesised image from the new video image shows the difference error image

where I is the vector of grey-level values in the image, and I_s is the vector of synthesized grey-level values for the current parameters. To locate the best match between model and

image, the aim is to minimize the squared magnitude of the difference vector, $\Delta = |\delta I|^2$, by varying the model parameters $\mathbf{f}$ (see Figure 3.13 and 3.8). The relationship between δI and the error in the model parameters is assumed to be linear :

$$\delta \mathbf{f} = A \delta I \tag{3.4}$$

To find A, multivariate linear regression is performed on a random sample of model displacements $\{\delta \mathbf{f}\}$ and the corresponding difference images $\{\delta I\}$.

Figure 3.13: Varying the first three parameters of the face model

The tracking method can be seen as an optimization problem. The best match for each frame must be found. The algorithm is as follows :

for each frame t of the sequence :

1. Evaluate the error vector δI_t with f_{t-1} the facial parameters from the previous frame.
2. Evaluate the current error Δ_t
3. Compute the predicted displacement $\delta f_t = A\delta I_t$
4. Set $k = 1$
5. Let $f_t = f_{t-1}.k\delta f_t$
6. Sample the image and calculate new error Δ'_t
7. If $\Delta'_t \leq \Delta_t$ then accept the estimate f_t
8. Else try with $k \in [1.5, 0.5, 0.25, 0.125]$ from step 5.
9. If $\Delta'_t > \Delta_t$ for all values of k then $f_t = f_{t-1}$ (remain with previous facial parameters)

From Cootes [17]

A face matching result when initialising a tracker using this algorithm is illustrated in Figure 3.14. After the initialisation the face will 'stick' to the face for the whole sequence.

original t=0 t=3 t=6 t=10

Figure 3.14: Evolution of fitting a face to the real image : the algorithm starts with a face at random position at instant $t = 0$. Evolution can be seen at $t = 3$, $t = 6$ and $t = 10$

We are now able to track a moving head by finding the closest match to a face generated by the face model. The resulting vectors contain the affine parameters and the combined shape/grey-level encoding the sequence seen.

3.2 The Speech Analyser

The speech is processed in two different ways, depending on the application (see Figure 3.15). The first way is to extract and store only the energy of the audio signal. An output signal is obtained, which can be used for future processing but cannot be used for regeneration. This method does not need any pre-learnt model. The second way is to extract audio parameters using a preprocessed voice model, as with the face tracker. These parameters describing the waveform can be stored and reused to recreate the speech.

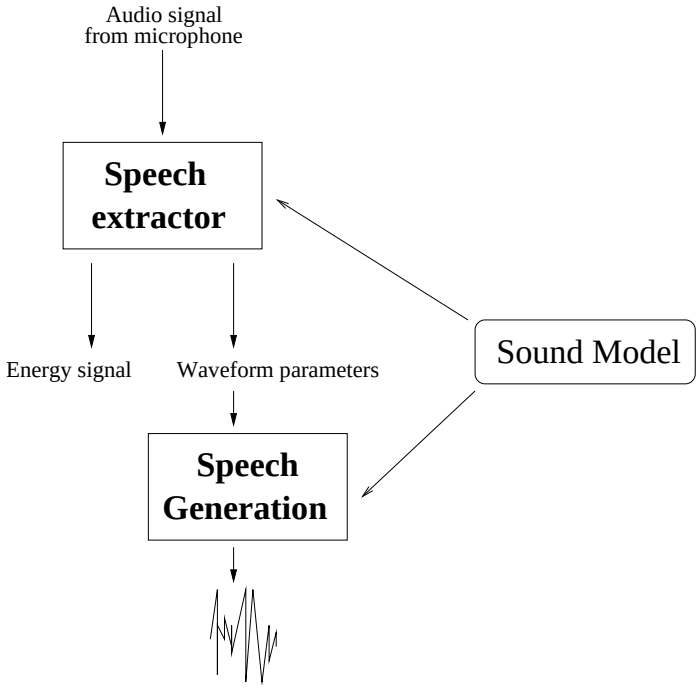


Figure 3.15: Sound acquisition

The sound acquisition is performed via microphones on headsets (see Figure 3.16) positioned low enough so no part of the face is hidden from the video camera.

In our experiments, the raw sound is sampled at 44.1KHz. This signal is partitioned into blocks of 4096 samples giving 10.76 blocks/second. This corresponds to the video frame rate used. Thus a one-to-one correspondence between audio blocks and video frames exists (see Figure 3.17).

Figure 3.16: Headset microphone

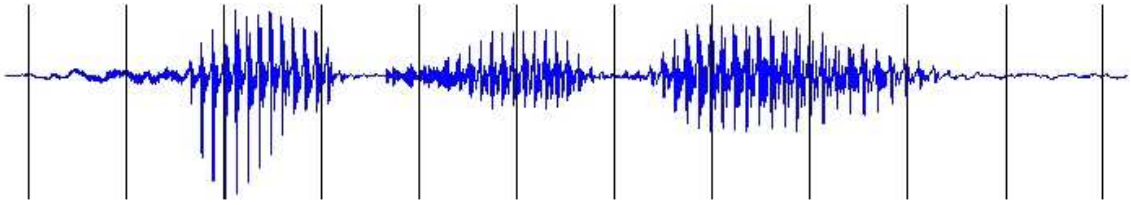


Figure 3.17: Signal cut every 4096 samples which correspond to every video frame

From the segmented signal we then have the choice between extracting the energy or extracting the speech parameters.

3.2.1 Energy of the voice

The energy signal is one of the simplest and most general piecess of information that can be extracted from an audio signal. The use of such a measure and examples for the datasets used in the thesis are shown in Chapter 5.

The equation for the energy calculation over a window $\{x_1, ..., x_w\}$ is :

$$E = \frac{\sum_{i=1}^{w} |x_i|}{w} \tag{3.5}$$

where

$$x = \text{signal}$$
$$w = \text{size of the energy window}$$

In our case the energy window, $w = 4096$, which is the same as the number of samples in each block. Figure 3.18 shows the energy signal extracted from a simple sentence.

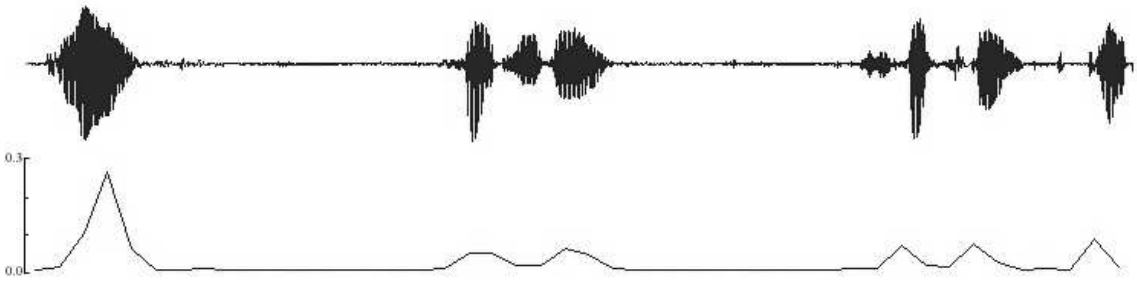


Figure 3.18: On the top is the waveform 'Hello, how do you do, do you fancy going to the pub?', on the bottom the energy graph

After the energy is extracted, the speech signal is lost and can not be regenerated. This is a one-way process only.

3.2.2 Waveform encoder

The purpose of the waveform encoder is to extract, against a preprocessed voice model, the corresponding values for the speech parameters. These parameters can be stored and reused to recreate the waveform (see Figure 3.19). To encode the frames using low-dimensional vectors, a Fourier Transform followed by a Principal Component Analysis is performed. All the processing is done from the audio signal of the person speaking.

In our experiments, the speech signal is sampled at 44.1kHz, giving 10.76 frames per second. A spectral analysis is performed on each frame using a Fast Fourier Transform (FFT) [90].

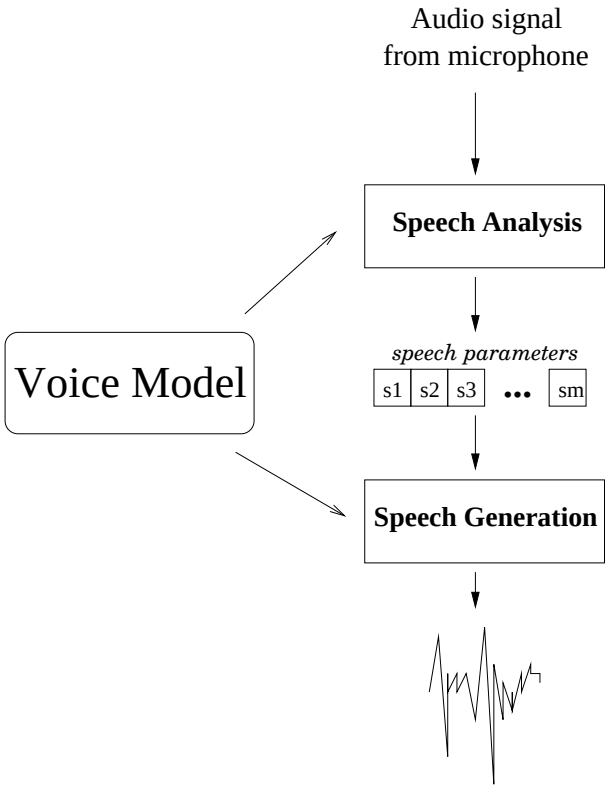


Figure 3.19: Waveform acquisition

The discrete Fast Fourier Transform is :

$$y_p = \sum_{k=0}^{n-1} x_k \left(cos\left(2\pi \frac{kp}{n}\right) + isin\left(2\pi \frac{kp}{n}\right)\right) \tag{3.6}$$

where

$$\begin{aligned} x_k &\text{ is the kth valued input sample} \\ y_p &\text{ is the pth complex-valued output} \\ n &= 2^N \end{aligned} \tag{3.7}$$

In our case $N = 12$ so $n = 4096$.

Applying principal component analysis (Section 3.1) to the vectors of spectral components from a training set of utterances allows a concise representation of individual frames (see Figure 3.20).

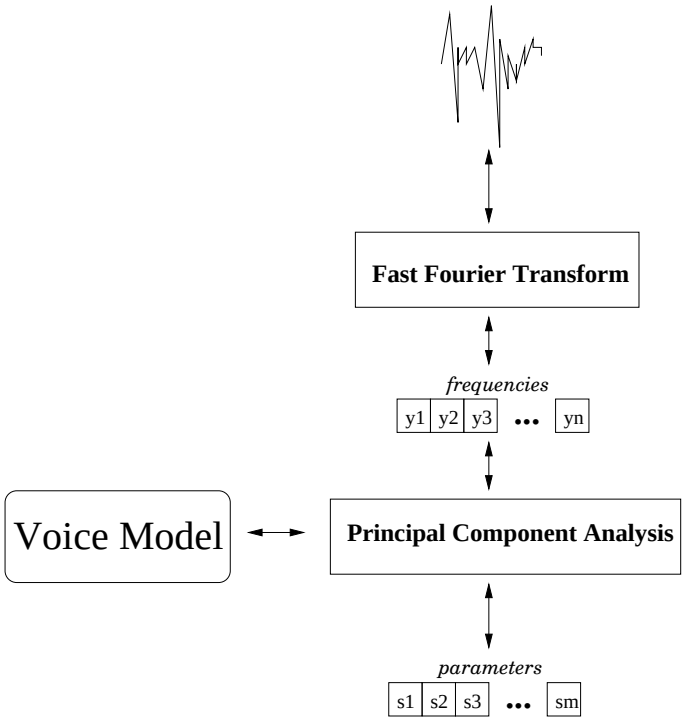


Figure 3.20: The encoding of the waveform

Any vector y in the frequency domain can then be approximated with :

$$y = \bar{y} + P_f s \tag{3.8}$$

where $\bar{y}$ is the mean, P_f a matrix with columns which are the principal orthogonal modes of variation and s a vector of speech parameters.

For our experiments, the first 70 principal components were found to give adequate reconstruction of the waveform (see Figures 3.21 and 3.22), accounting for about 85% of the variance. Thus, each frame (4096 waveform samples, lasting 93ms) is represented by 70 parameters. In general, a frame of sound is represented by assigning values to the n parameters of the voice model: $\mathbf{s} = \{s_1, s_2, ..., s_n\}$

In our experiments the voice model was able to reproduce with fidelity all of the vocabulary occurring in the utterances it had been trained with. Any utterance not occurring in the training set would be expected to give a poor reconstitution, like that shown on the bottom-left of figure 3.22. The utterances used for the training set can be seen in

Chapter 5, Table 5.2. In figure 3.21, the residual error for different utterances with the use of different numbers of principal components can be observed. Note that the French 'bonjour' is relatively well matched globally (i.e. a comparable residual) even though it was not part of the training set. We can observe in Figure 3.22 that the specific French pronunciation for the 'j' is not reconstituted as it was not in the training set.

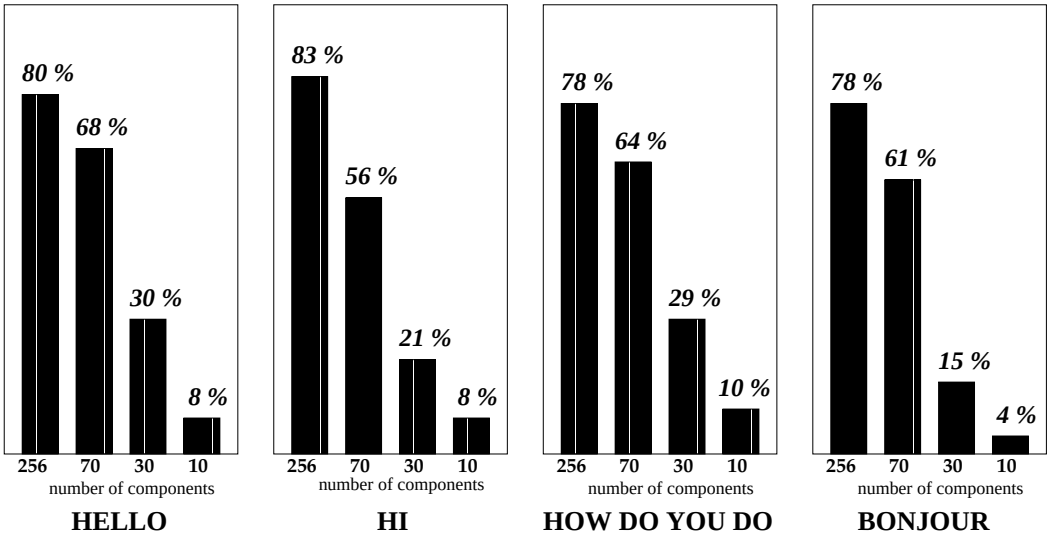


Figure 3.21: Quality of reconstitution of different utterances with the use of different numbers of components. Only 'Bonjour' did not occur in the training set.

Finally, the two vectors, representing video and sound, containing the parameters describing facial expression and speech or energy signal are combined. The face parameters are appended to the speech/energy parameters and will be used as one single vector for the behaviour learning process. Figure 3.23 illustrates this process.

This chapter has discussed the data acquisition and processes for encoding the video and audio data. The final joint vectors obtained are stored and can be used to redisplay the video and sound encoded or processed by our behaviour system. The behaviour system allowing an interactive response from a virtual talking head is introduced in the following chapters.

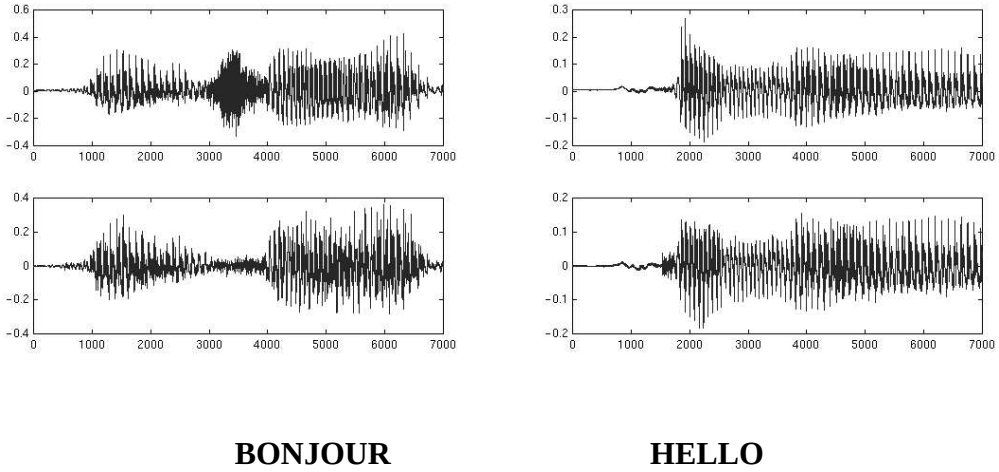


Figure 3.22: On the left the sound 'bonjour' (top) with reconstitution (bottom). This sound is not in the training set. On the right the sound 'Hello' (top) with reconstitution (bottom)

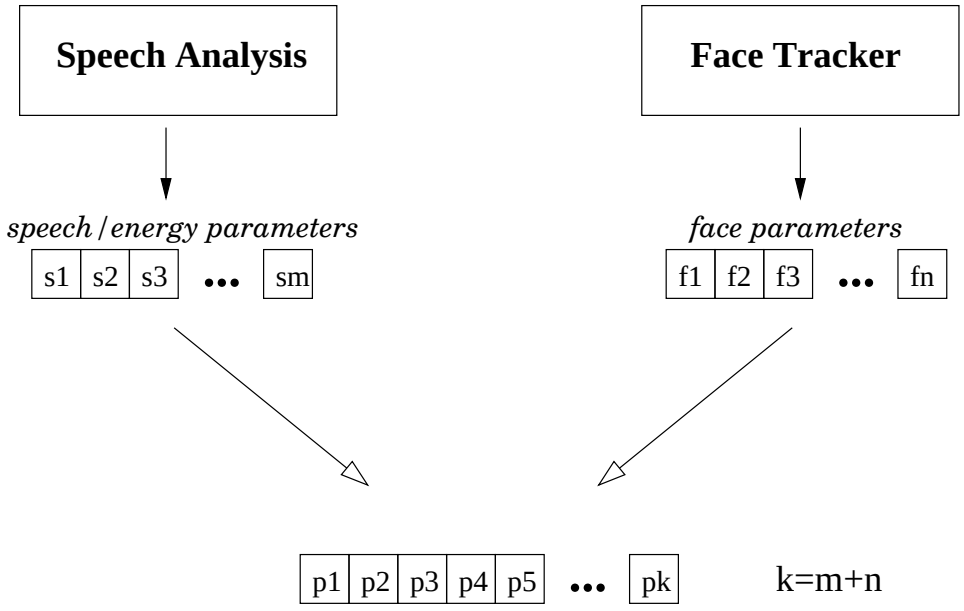


Figure 3.23: Video encoded vector and speech encoded vector are combined

Chapter 4

Modeling Interaction

This chapter describes a method for automatically learning models of object interaction [51] and how this method is used to learn models of interactive behaviour between two individuals in the case of face to face conversation. The learning system initially generates a set of *state* prototype vectors describing the configuration space (facial expression and sound). State prototype vectors together with a history-trace memory mechanism are then used to generate *behaviour* prototype vectors describing the interactive behaviour space.

In this thesis we are interested in modelling the joint behaviour of two interacting objects. In order to encode interaction between two people we use a joint model of the two talking faces. The vectors encoding face and speech for each talker are joined into a single one as illustrated in Figure 4.1.

4.1 The joint model

The two interacting talking heads, T^A and T^B, combining speech and facial movement are joined into single vectors T.

The vectors are encoded as:

$$T = \{T^A, T^B\} \tag{4.1}$$

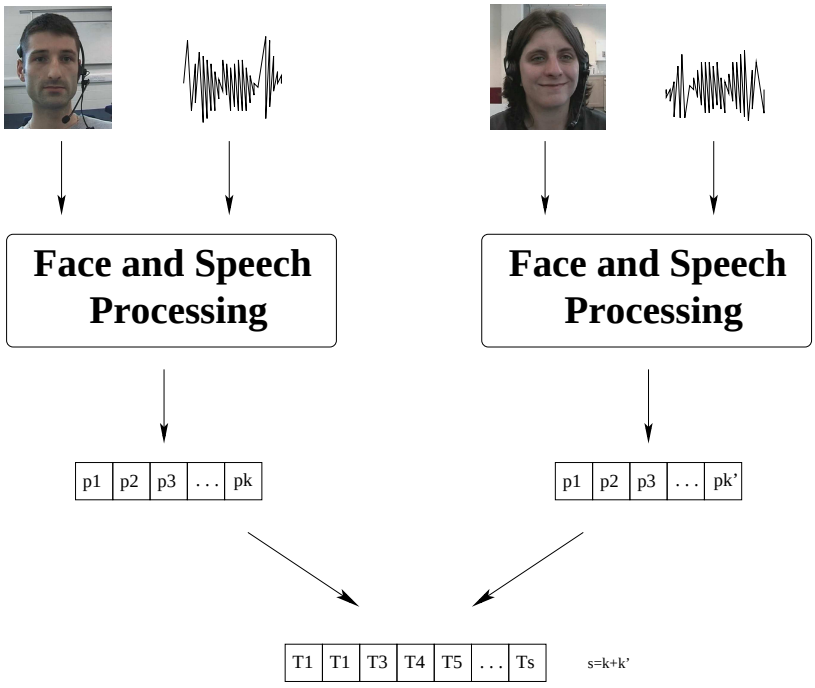


Figure 4.1: Joint vector of the combined face/speech vectors for the two talking-heads

where

$$T^A = (p_1, p_2..., p_m)$$
$$T^B = (p_1, p_2..., p_n)$$

Prior to modelling, the joint vectors are scaled and translated to lie within a unit hypercube. This is to simplify future stages of the encoding process. These normalised vectors are configuration vectors C.

An animated facial expression with speech is now represented by a sequence of configuration vectors $C_t \in [0,1]^s$:

$$\{C_1, C_2, ..., C_l\} \tag{4.2}$$

with l the length of the animated talking heads sequence.

All the vectors C_t within the hypercube $[0,1]^s$ are a conjunction of the two talkers (normalised talking head A on the right and talking head B on the left). To illustrate, a simple

interaction within the hypercube of two people greeting each other shown in Figure 4.2 is plotted in Figure 4.3.

Figure 4.2: Two people greeting each other. On first row speaker one saying 'Hello' and on second row speaker two answering 'Hello'

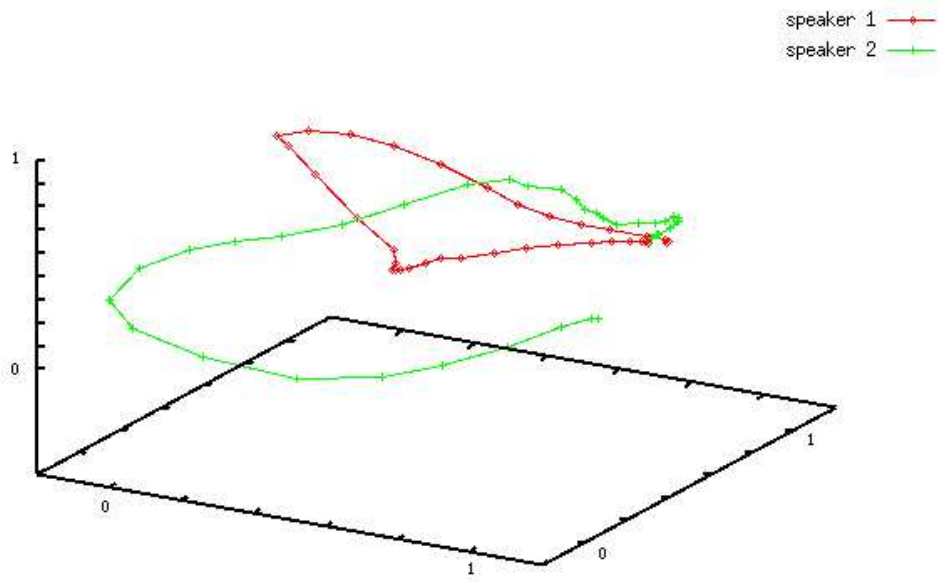


Figure 4.3: The first three parameters of the combined model (greylevel+shape) of each speaker saying 'hello' within the hypercube

4.2 The learning system

From the face and speech database acquired in a previous learning phase a reduced space is automatically created. From that new space and some interaction clips we obtain a time-based space composed of face and speech information. This time-based space will be once again reduced to a smaller space and a Markov chain will be superimposed to represent these interaction behaviours. The schema on Figure 4.4 summarize the process.

The representation of behaviours and the associated learning mechanism is essentially that proposed by Johnson *et al.* [51, 50].

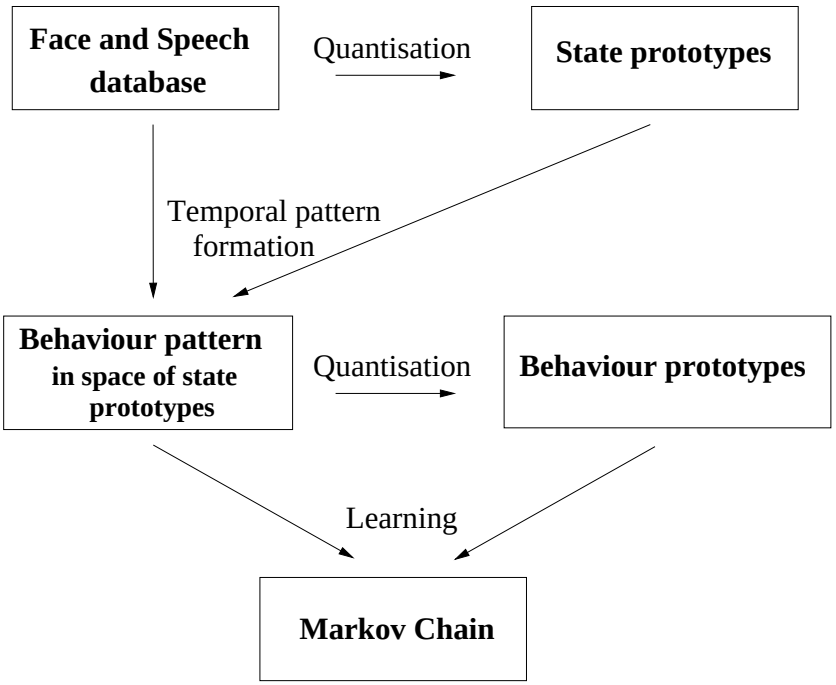


Figure 4.4: Learning system

4.2.1 Learning configuration space

To encode the behaviour clips we use sequences of *state vectors* representing the evolution of the configuration and the first derivative of configuration.

State vectors $F_t \in [0,1]^{2s}$ consist of a configuration vector C_t and its derivative $\dot{C}_t$. To produce training data, the first derivative is approximated by the difference in configuration vector between successive frames:

$$F_t = (C_t, \lambda \dot{C}_t + H) \tag{4.3}$$

where

$$\dot{C}_t = C_t - C_{t-1}$$

λ is a scaling factor and $H \in \Re^s$ is a vector with all components equal to $\frac{1}{2}$.

H and λ are chosen so components of state vectors span the interval $[0,1]$ (for more details see [50]).

The evolving behaviour of a talking face is represented by sequences of state vectors of the form :

$$\mathcal{F} = \{F_1, F_2, ..., F_l\} \tag{4.4}$$

To create a model of behaviour we use a training set of observed sequences $\mathcal{O} = \{\mathcal{F}_1, \mathcal{F}_2 ..., \mathcal{F}_o\}$

To make the behaviour training set invariant to the starting position in translation, rotation and scale, all the behaviour clips $\mathcal{F}$ are expressed relative to the 4 affine components X_1, Y_1, R_1 and S_1 (see previous chapter) of their first frame F_1 :

For each $F_t \in \mathcal{F} \in \mathcal{O}$

$$X'_t = X_t - X_1$$
$$Y'_t = Y_t - Y_1$$
$$R'_t = R_t - R_1$$
$$S'_t = S_t - S_1$$

Frame 1 for each $\mathcal{F}$ is thereby transformed to $X'_1 = Y'_1 = R'_1 = S'_1 = 0$.

The distribution of all the vectors F of $\mathcal{F} \in \mathcal{O}$ is modelled using a vector quantisation [35]. This allows any distribution to be characterised by a fixed number of prototype state vectors (see Figure 4.5).

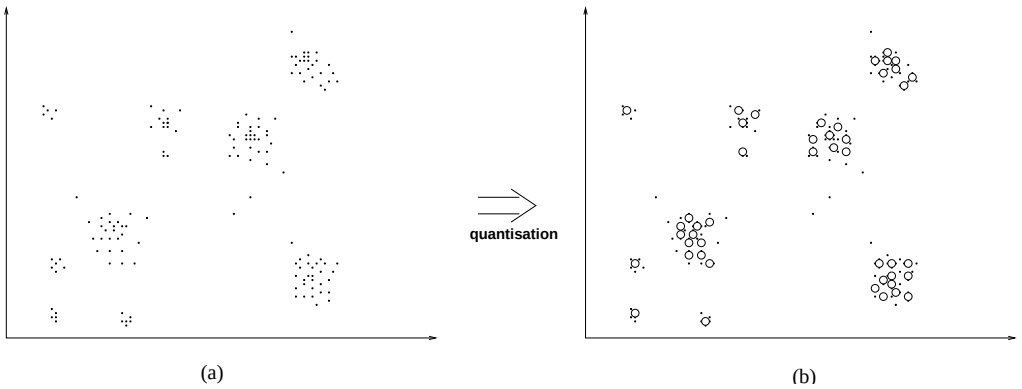


Figure 4.5: (a) depicts a set of vectors in the state space, (b) shows the same set of vectors with prototypes superimposed

The final distribution of the prototypes by a regular vector quantisation is extremely sensitive to their initial placement within the feature space. To solve that problem we incorporate a prototype sensitivity mechanism developed by Johnson [51] and inspired from Bienenstock *et al.* [5]. Each prototype can automatically vary their sensitivity to exclude themselves from the competition. This variation of the Vector Quantisation is called *Altruistic Vector Quantisation* (AVQ) [51].

Each prototype is associated with a sensitivity value $S_i(t)$ which modifies the Euclidean distance metric normally used in competitive learning. A prototype with a positive value is more likely to win the competition than a prototype with a negative value.

The complete algorithm, consolidated from Johnson [50] is on the next page :

The algorithm places a set of m prototypes $c_i \in [0,1]^{2s}$, referred to as a **codebook**, over N iterations:

1. Randomly place the m prototypes within the unit hypercube $[0,1]^{2s}$.
2. Select $\mathbf{z}(t)$, the current training vector, randomly from the distribution.
3. Find the prototype $\mathbf{c}_j(t)$ which is nearest to the current training vector $\mathbf{z}(t)$:

$$|\mathbf{z}(t) - \mathbf{c}_j(t)| - S_j(t) = min_i|\mathbf{z}(t) - \mathbf{c}_i(t)| - S_i(t). \tag{4.5}$$

where $S_i(0) = 0$ and sensitivity values are updated on each iteration using

$$S_i(t+1) = \zeta S_i(t) + A_i \tag{4.6}$$

where ζ is a *damping coefficient* defined as

$$\zeta = 1 - \frac{\beta}{(k-1)\sqrt{d}}, \tag{4.7}$$

and A_i introduces sensitivity adjustments defined by

$$A_i = \begin{cases} -\beta & \text{if } i = j \\ \frac{\beta}{k-1} & \text{otherwise,} \end{cases} \tag{4.8}$$

where β is a constant in the interval $(0,1)$ specifying the magnitude of adjustments.

4. Update prototypes as follows:

$$\mathbf{c}_i(t+1) = \begin{cases} \mathbf{c}_i(t) + \alpha(t)[\mathbf{z}(t) - \mathbf{c}_i(t)] & \text{if } i = j \\ \mathbf{c}_i(t) & \text{otherwise,} \end{cases} \tag{4.9}$$

where $\alpha(t)$ is a monotonically non-increasing gain coefficient,

$$\alpha(t) = \begin{cases} 1 - 0.99(\frac{2t}{N}) & \text{if } 0 \leq t < \frac{N}{2} \\ 0.01 & \text{if } t \geq \frac{N}{2} \end{cases} \tag{4.10}$$

referred to as the *cooling schedule* of the learning process

5. Repeat steps 2-4 for N iterations.

Adapted from Johnson [50]

The resulting state models after the vector quantisation are a set of u state prototypes $\bar{\alpha}_i$:

$$\mathcal{A} = \bar{\alpha}_1, \bar{\alpha}_2, ..., \bar{\alpha}_u \tag{4.11}$$

Those prototypes will be used as an alphabet to encode the behaviours.

Figure 4.6 shows the first 200 joint vector prototypes generated after the AVQ on the configuration space. The training set utterances can be seen in Chapter 5, Table 5.2.

4.2.2 Learning behaviour space

To encode a behaviour, we must encode paths through state space. Each behaviour can then be encoded as a trajectory in the state space and projected in the spatio-temporal space to be processed. Those trajectories have to be compared to each other to decide how similar they are. To encode those trajectories we use a memory mechanism to keep a history of the proximity to each state prototype. We then model the distribution of those behaviours using the Altruistic Vector Quantisation (see Figure 4.7)

The memory mechanism approach is similar to the Leaky Integrators of Reiss and Taylor [85] and the Neurons of Wang and Arbib [107].

A real valued 'activation' level z is associated with each state prototype. The activation of a prototype gives a measure of the elapsed time since the last time it was close by slowly decaying over a period of time. The whole set of prototypes forms a trace of the history which we use as an encoding of the trajectory.

Using the prototypes $\mathcal{A}$ as alphabet, the temporal pattern formation is encoded in the proximity of successive state vectors from an ordered set $\mathcal{F}$ to the corresponding prototype $\bar{\alpha}$. The proximity $p_i(t)$ of a state vector F_t to a state prototype $\bar{\alpha}_i$ decreases linearly from one to zero as the distance between them increases from zero to the maximum separation within the hypercube state space:

$$p_i(t) = 1 - \rho \left(\frac{F_t - \alpha_i}{\sqrt{2s}} \right), \tag{4.12}$$

where $2s$ is the dimensionality of the state space and ρ is a scaling factor chosen such that

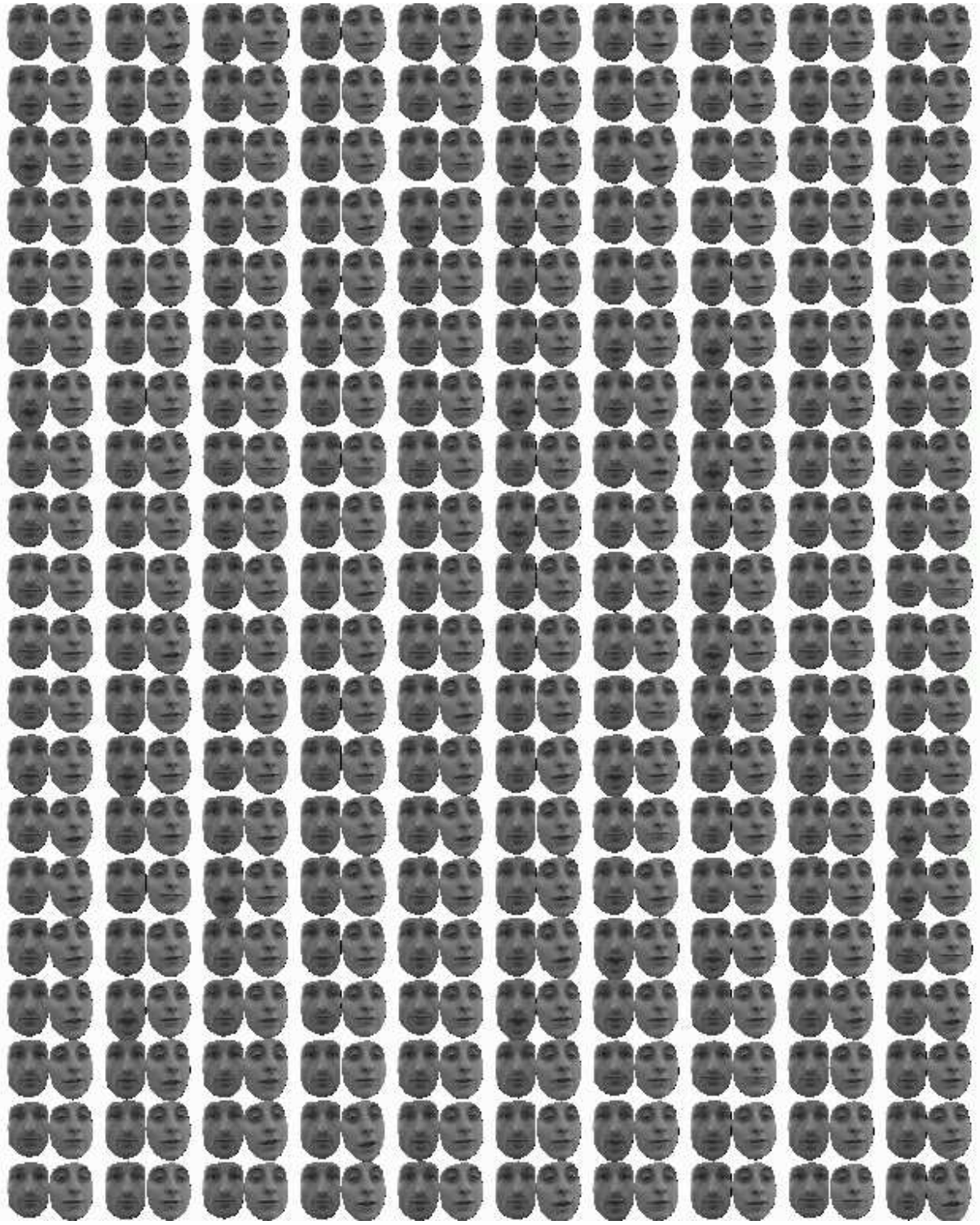


Figure 4.6: Prototypes $\bar{\alpha}$ of the joint configuration space

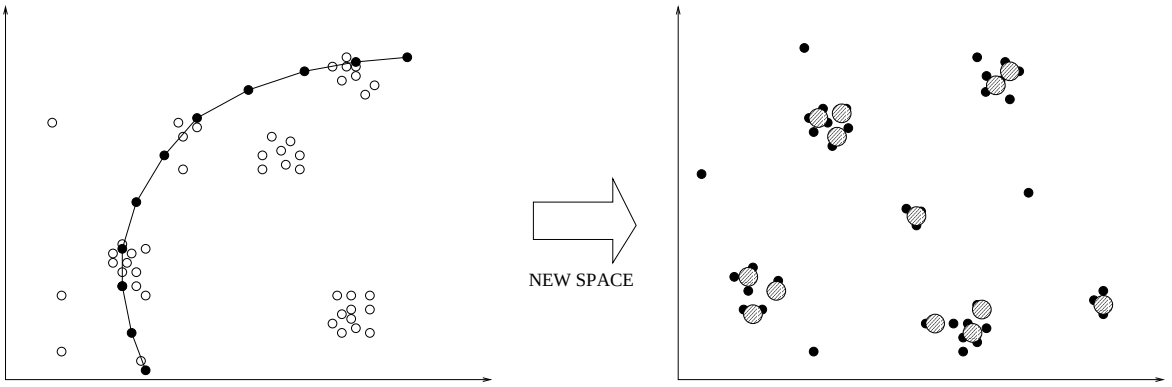


Figure 4.7: Distribution of the prototypes in the trajectory space

$\frac{\sqrt{2s}}{\rho}$ is approximately equal to the maximum observed separation within state space. The activation level of every prototype is updated at each iteration.

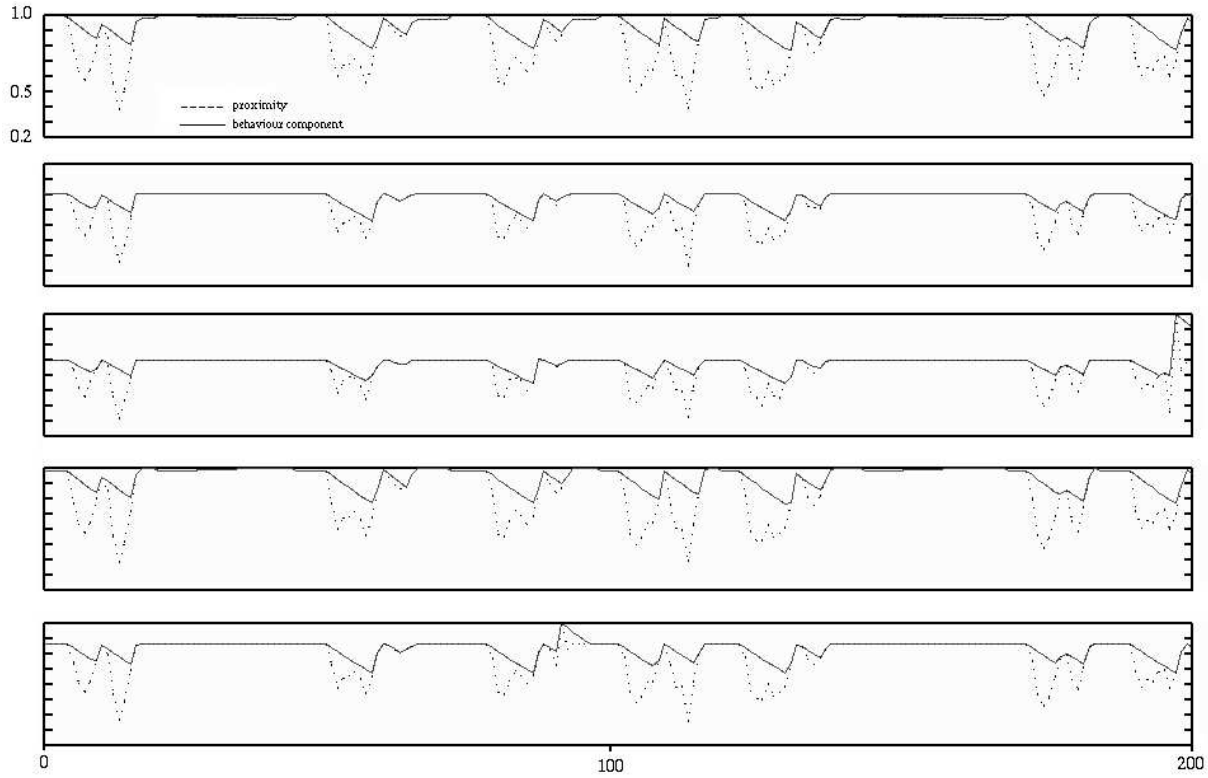


Figure 4.8: Conditional decay operator applied to sample proximity data with $\gamma = 0.97$

We apply a conditional decay operator to these proximity values to retain a trace of the

history :

$$z_i(t) = \begin{cases} p_i(t) & \text{if } p_i(t) > \gamma z_i(t-1) \\ \gamma z_i(t-1) & \text{otherwise,} \end{cases} \tag{4.13}$$

where γ is a coefficient in the interval $[0, 1]$ which defines the rate of decay. For all reported experiments, γ is set to 0.97. On the graph in Figure 4.8 we can see the conditional decay operator applied over time to the four first tokens of the alphabet (proximity on vertical axis)

The 'behaviour' vector Z_t is formed from the set of $z_i(t)$ values.

$$Z_t = (z_1(t), z_2(t), ..., z_u(t)) \tag{4.14}$$

where u is the cardinality of the state prototype set.

The sequenced set of behaviour vectors $\{Z_t \in [0, 1]^u\}$ is generated from the set $\mathcal{A}$ of u prototypes and the l state vectors F_t:

$$\mathcal{Z} = Z_1, Z_2, ..., Z_l \tag{4.15}$$

Using the AVQ algorithm with the training sets $\mathcal{Z}_j$ we obtain a set of v behaviour prototypes $\bar{\beta}_i$:

$$\mathcal{B} = \bar{\beta}_1, \bar{\beta}_2, ..., \bar{\beta}_v \tag{4.16}$$

4.2.3 Markov Chain

Unfortunately, the behaviour model represented by the learned behaviour prototypes can not be easily used for generative purposes, that is for behaviour synthesis or prediction. In order to facilitate the generative tasks using the learned model, a Markov Chain $\mathcal{M}$ (see Figure 4.9) superimposed on the set of behaviour prototypes $\mathcal{B}$ is defined by the 4-tuple

$$\mathcal{M} = (\mathcal{E}, \mathcal{S}, \pi, \mathcal{T}) \tag{4.17}$$

where

$$\mathcal{E} = \{e_1, e_2, ..., e_{k+1}\} \tag{4.18}$$

is the set of chain states, with e_{k+1} the *end state*,

$$\mathcal{S} = \{\bar{\beta}(e_1), \bar{\beta}(e_2), ..., \bar{\beta}(e_{k+1})\} \tag{4.19}$$

is the set of state vector tokens associated with the chain states,

$$\pi = \{\ \pi_1, \pi_2, ..., \pi_k\}, \pi_i = P(e_i \text{ at step } r = o) \tag{4.20}$$

defines the initial state distribution, and finally,

$$\mathcal{T} = \begin{bmatrix} \mathcal{T}_{1,1} & \dots & \mathcal{T}_{1,k+1} \\ \vdots & \ddots & \vdots \\ \mathcal{T}_{k,1} & \dots & \mathcal{T}_{k,k+1} \end{bmatrix}, \mathcal{T}_{i,j} = P(e_j \text{ at step } r+1 | e_i \text{ at step } r) \tag{4.21}$$

is a matrix defining the state transition distribution.

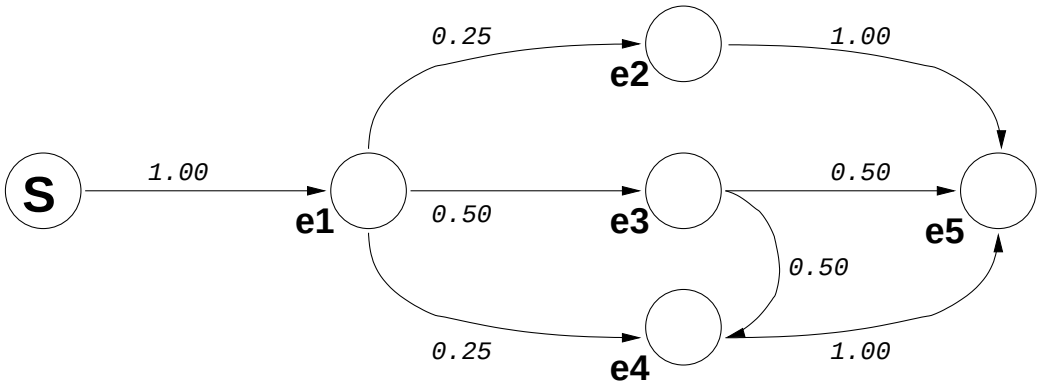


Figure 4.9: Example of Markov Chain superimposed on a set of six prototypes

The initial state distribution and state transition matrix are estimated from the sequences of behaviour prototypes derived from the training set. An example of the transition matrix built over a 30 prototypes alphabet can be seen in Figure 4.10. This example was built over a relatively small alphabet and was based on the 'listening head' application described in the next chapter.

4.2.3.1 Optimisation in Markov Chain generation

Although behaviour prototypes encode sequences of state prototypes it is not possible to reconstruct an approximation of these sequences. Sequenced behaviour prototypes are needed to reconstruct a state sequence.

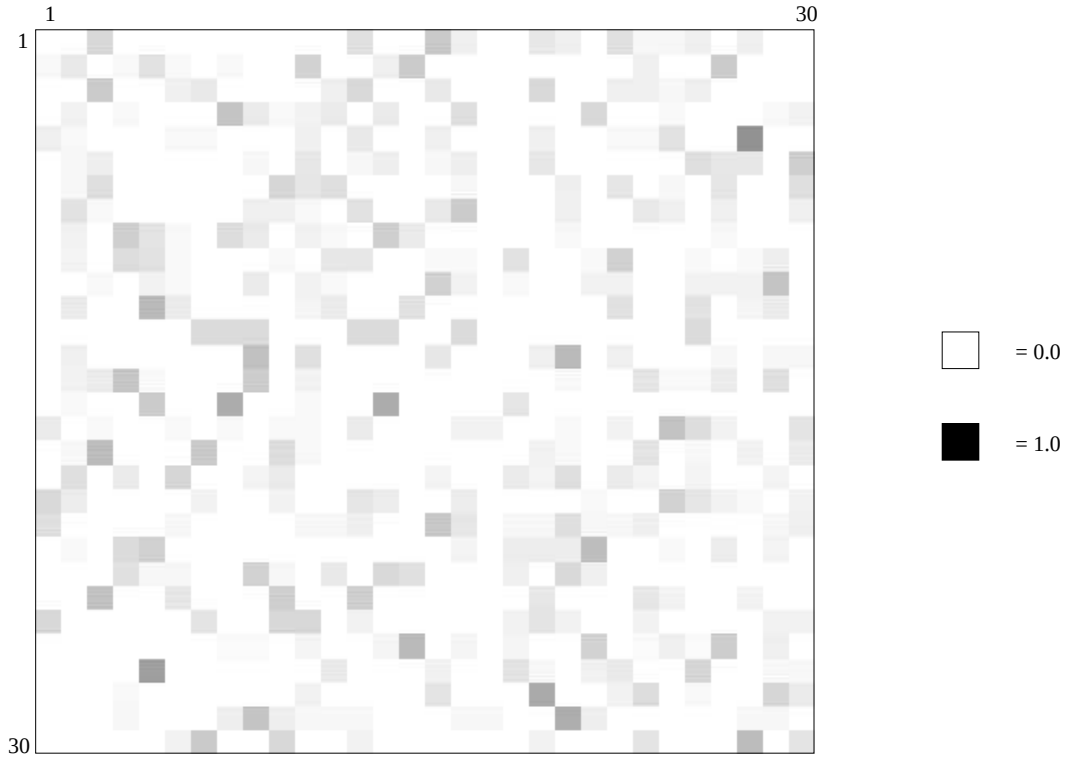


Figure 4.10: Values of the transition of the 20 states Markov chain built for the listening head

The highest valued component of a behaviour prototype might be expected to be a good way to select the state prototype at the end of a given behaviour, see Johnson [50]. Note this is rather poor compared to the mean state of the behaviours represented by each prototype. The same prototype is often associated with a sequential chain state and we end up with identical adjacent state vectors. It can be seen as a loss of detail if we try to predict or regenerate an interaction. To optimize the travelling of the Markov chain and eliminate these spatio-temporal inaccuracies the behaviour prototype is replaced, during the learning, with the actual mean current state of the behaviour represented by each of the behaviour prototypes :

$$\bar{\beta}(e_i) = \frac{\sum_{j=1}^{n} F_j}{n} \tag{4.22}$$

where F_j are the current state vectors associated with the behaviour represented by prototype $\bar{\beta}_i$

We're now able to encode behaviour vectors in the form of trajectories and apply a Markov

chain on top of it with the transitions automatically learnt. The following section will describe how we use joint behaviour vectors to encode interactions.

In the next chapter the use of this model to generate interaction between speakers and the interactive talking head will be described and demonstrated. Other uses of that model and other possible applications of a talking head will be discussed.

Chapter 5

The Talking Head

This chapter is concerned with the use of an interaction model to drive a synthetic talking head.

The interaction model has been generated by the learning process (see previous chapter) from observations of interaction between two people. From a new observation of only one individual I generate an appropriate response using the model. The observation and generation of facial expression and speech are obtained via a face and speech model (see Chapter 3). In Figure 5.1 a summary of the learning process and the approach I use to generate a response from a new input can be seen.

The first section of this chapter will describe this approach in detail. Then a first application of a generated talking head with results will be shown and discussed. Three other applications are then presented based on variations of the process, using a different kind of input and output.

The four applications described and evaluated in this chapter are :

- Application I : the talking head. A synthetic talking head replying to greetings, in Section 5.2.
- Application II : the listening head. A synthetic talking head which acknowledges the utterances of a speaker, in Section 5.3.

- Application III : assisting a partially sighted listener. A speaking assistant which describe vocally facial expressions, in Section 5.4.
- Application IV : a behaviour filter. A filter which detects atypical behaviours, in Section 5.5.

We will finish with an evaluation of the interaction quality of the talking head.

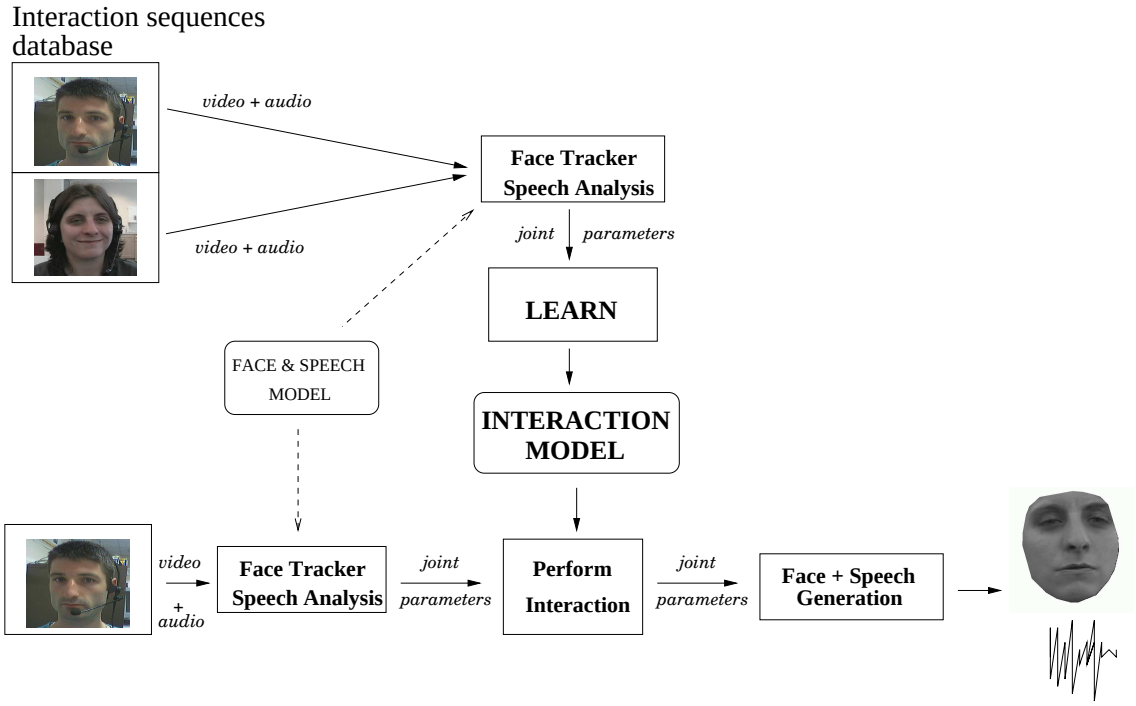


Figure 5.1: Talking Head learning system and application

5.1 Generating a synthetic response

After the learning process described in Chapter 4, we end up with a Markov chain built over a set of behaviour vectors. Those vectors and the matrix $\mathcal{T}$ of state transition probabilities are learnt from a training set (see Chapter 4).

Travelling through the Markov chain starting from any start state and respecting the transition distribution until reaching an end state will generate an interaction between

two speakers. The following section describes how the model is used to respond to real talking heads.

Each state of the Markov chain is a state vector combining two talking heads $F = T^A, T^B$. We are going to use only one talking head given as input T^A and the other part of the state vector will be generated as a synthetic talking head T^B.

In tracking the user during an interaction, we must deal with uncertainty in the elements of that part of the state vector acquired from pre-processing each incoming frame. Following Johnson *et al.* [51], a Bayesian framework is adopted in which the posterior density for a hypothesised state F_t at each time-step is estimated recursively from a prior density for the state and a likelihood function given the current observation T_t^A:

$$P(F_t|T_t^A, ..., T_0^A) \propto P(T_t^A|F_t)P(F_t|T_{t-1}^A, ..., T_0^A) \tag{5.1}$$

where $P(F_t|T_t^A, ..., T_0^A)$ is the conditional distribution of state given an observation history, $P(T_t^A|F_t)$ measures the *likelihood* of a state F_t giving rise to observation T_t^A, and $P(F_t|T_{t-1}^A, ...T_0^A)$ is the *prior* distribution representing predictions from the *posterior* distribution $P(F_{t-1}|T_{t-1}^A, ...T_0^A)$, from the previous time step.

A Gaussian likelihood function is used, based on the error relating to hypothesis F_t^A $E(F_t^A, T_t^A)$:

$$P(T_t^A|F_t) = exp\left(-\frac{E(F_t^A, T_t^A)^2}{2\sigma^2}\right). \tag{5.2}$$

where σ, in our case, is chosen to 0.05. The hypothesis error is based on the Euclidean distance between the observed vector T_t^A and the hypothesis states of the Markov Chain F_t.

$$E(F_t^A, T_t^A) = |F_t^A - T_t^A| \tag{5.3}$$

The CONDENSATION tracking algorithm of Isard and Blake [46], in which the posterior

density is represented by a set of sample hypotheses, is applied. In our experiments, a total of 100 sample hypotheses were found to be adequate.

The algorithm, from Johnson *et al.* [51], is as follows :

1. Generate a set $\mathcal{X}_0$ of N hypotheses to represent the initial prior, where $\mathcal{X}_0$ is obtained by sampling with replacement from the initial state distribution π.
2. For each $F \in \mathcal{X}_t$, use the error $E(F^A, T_t^A))$ to calculate the likelihood of the hypothesis using Equation 5.2.
3. Use relative likelihood values to weight sampling from $\mathcal{X}_t$, the prior, resulting in a set $\mathcal{Y}_t$ of N hypotheses representing the posterior distribution.
4. Produce the virtual response T_t^B from the hypothesis in $\mathcal{Y}_t$ with maximum likelihood.
5. Generate a new set $\mathcal{X}_{t+1}$ of N hypotheses to represent the new prior, where each $F' \in \mathcal{X}_{t+1}$ is a stochastic extrapolation at time $t+1$ from $F \in \mathcal{Y}_t$ using the transition probabilities $\mathcal{T}$.
6. Repeat steps 2-5 until the interaction is complete.

adapted from Johnson [51]

This algorithm generates a virtual talking head T_t^B from a single (a single head with a single voice) input T_t^A.

The following sections propose some applications of this method. Experiments and results are presented.

5.2 Application I : the talking head

In this first application we generate a synthetic head answering to simple greetings from a real head. The model is trained with video sequences of two people greeting each other where both audio and speech are processed.

The acquisition was staged. The two actors were facing each other, wearing a microphone headset and two cameras were pointing at them, one on each actor, see Figure 3.2. The first actor repeatedly greets the second who responds appropriately. After processing

Learning clips		
Question	Answer	Number
"Hello ?!"	"Hello !"	31
	TOTAL	31

Table 5.1: Learning clips in experiment 1 : different intonations, speed and timing

the sequences are manually segmented into interaction clips starting with a greeting and ending with the corresponding response.

Two experiments covering interesting aspects of this application are going to be presented with their respective results. A general evaluation of the quality of interaction of the talking head is proposed in Section 5.6

We performed two experiments in which a behaviour model is build with different kinds of interaction clips.

5.2.1 Experiments

In the first experiment the system is trained with 31 simple interaction clips involving the greetings 'Hello' associated with the natural 'hello' response. For later comparison the utterances used are shown in Table 5.1.

These interactions differ from one another in intonation, facial expression, speed and timing. Each interaction clip lasts between 2 and 2.5 seconds.

The face model for the first speaker had 14 components (see Section 3.1) and the speech 70 components (see Section 3.2). The face model for the second speaker had 10 components and the speech 70 components. The variation in the number of components for the face tracking is due to the first speaker's more complex facial appearance.

The 31 interaction clips used represent a total of 740 frames. The model was trained with 500 configuration prototypes and 500 behaviour prototypes. We end up with a Markov Chain of 379 connected states, we use 100 hypotheses for the propagation of a posterior

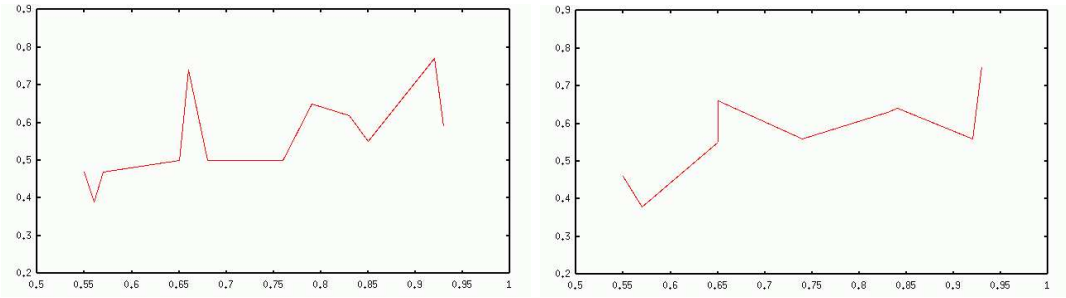
density.

The model is then tested with new 'hello' interaction clip as input. Each of these clips are an interaction between speaker one and two where all the parameters regarding speaker two are hidden.

The first experiment results illustrated in 5.3 show some different outputs.

In Figure 5.3(a) and Figure 5.3(b) we can observe a correct response with a little pause between the questions and the answers. This kind of timing was observed many times in the sequences used to build the model. It is also a very natural way to greet each other.

We can observe in Figure 5.3(a) and 5.3(b) that the speed of the response varies depending on the speed of the first speaker. The correlation between the length of the greeting of first and second speakers was observed in the training section. It has been modelled and the synthetic response respects it. In Figure 5.2, the corresponding time length between the greetings and the answers of the training set and the corresponding time length between new input greetings and their respective generated responses is shown. A slight upward trend from left to right can be observed.



(a) (b)

Figure 5.2: On the left (a) the real observation correlation timing between questions (horizontal axis) and answers (vertical axis). On the right (b) the observation correlation timing between real questions (horizontal axis) and synthetic answers (vertical axis). Units are in seconds.

In Figure 5.3(c) the timing is a bit unusual and was observed in some of the training clips.

Learning clips		
Question	Answer	Number
"Hello ?!"	"Hello !"	8
"Hi ?!"	"Hi !"	8
"How do you do ?"	"Fine !"	12
Smile	Smile	9
-	-	1
	TOTAL	40

Table 5.2: Learning clips: different interactions with different intonations

Clues in the facial expression, or the simple fact that the listener is expecting a greeting, allow him to start answering before the first speaker is finished. This result shows that the model includes those kind of variations in the timing and can generate fast and realistic responses which look natural. In the evaluation experiment in Section 5.6, the subjects reactions to that sequence are shown.

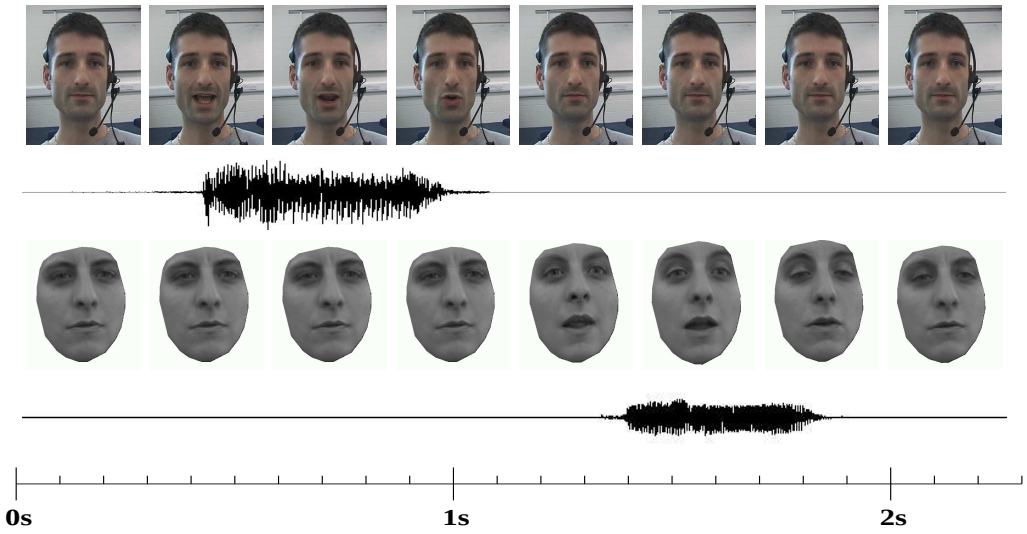
Experiment with several greetings:

In the second experiment the system is trained with 40 simple interactions involving the greetings 'Hello', 'Hi' and 'How do you do?', with associated responses. The utterances used are shown in Table 5.2.

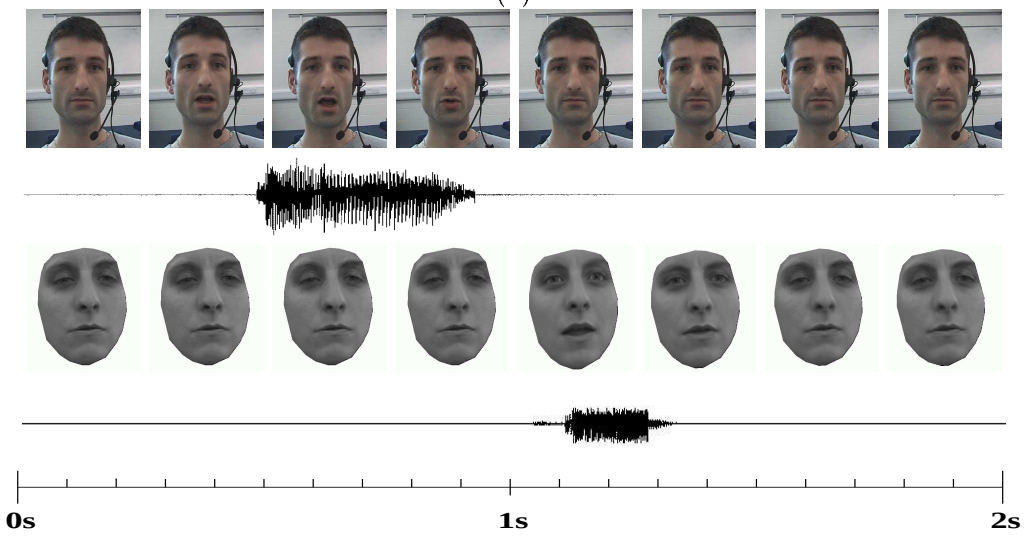
One interaction clip in which nothing happens is included. This will create some simple trajectories in the behaviours space of two static faces with no speech. The model will then be able to deal with a silent speaker. Each interaction clip lasts between 2 and 2.5 seconds.

The face model for the first speaker had 10 components (see Section 3.1) and the speech 70 components (see Section 3.2). The face model for the second speaker had 10 components and the speech 70 components. Again, the variation in the number of components for the face tracking is due to speaker one's more complex facial appearance.

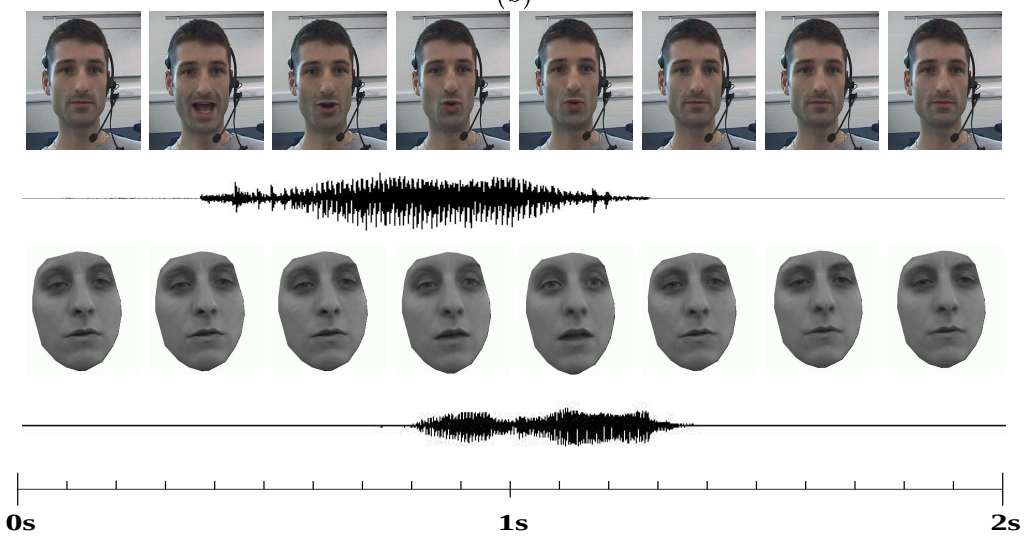
The 38 interaction clips used represent a total of 899 frames. The model was trained with



(a)



(b)



(c)

Figure 5.3: In (a), (b) and (c) : On first row the face as input saying "hello" with the associated speech waveform and on second row the synthesised face with the associated speech waveform.

600 configuration prototypes and 600 behaviour prototypes. We end up with a Markov Chain of 486 connected states using 100 hypotheses for the propagation.

The model is tested with new inputs 'how do you do?' and 'smile'. Then a weird input with unrecognisable speech (the audio signal 'How do you do' reversed while the face is smiling) is tested.

The results of the second experiment are illustrated in Figure 5.4 and Figure 5.5.

Figure 5.4(a) and Figure 5.4(b) show appropriate answers to new inputs with natural timing. The result in Figure 5.5 for the abnormal input shows how the model deals with something unusual.

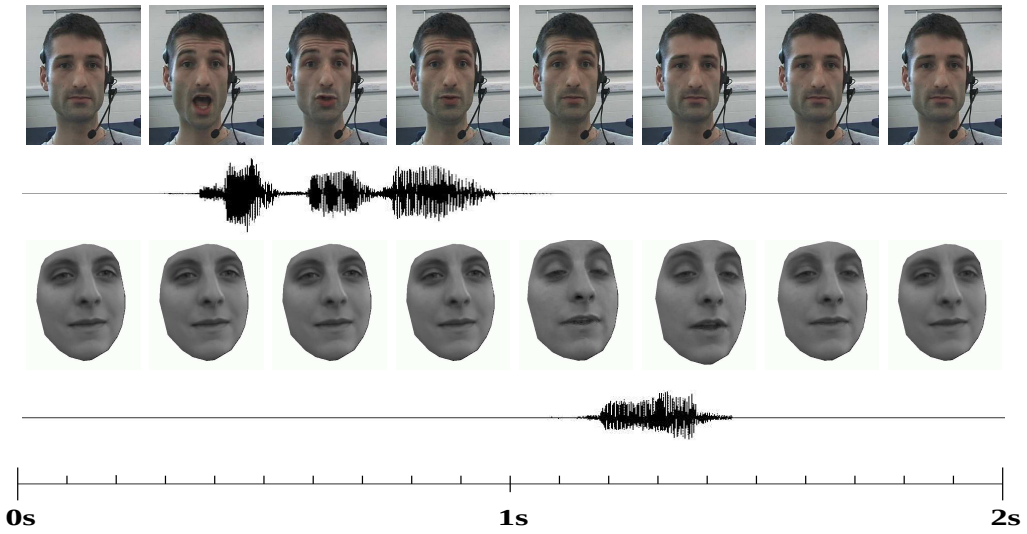
As the audio was a reversed signal of 'how do you do' and the facial expression associated a smile (mouth closed), none of the states of the Markov Chain can represent it. The algorithm keeps jumping randomly from state to state until it stabilises on a specific closest path, which in our case is a smile with silent audio.

More experiments on how the model reacts to unusual input can be seen in Section 5.5.

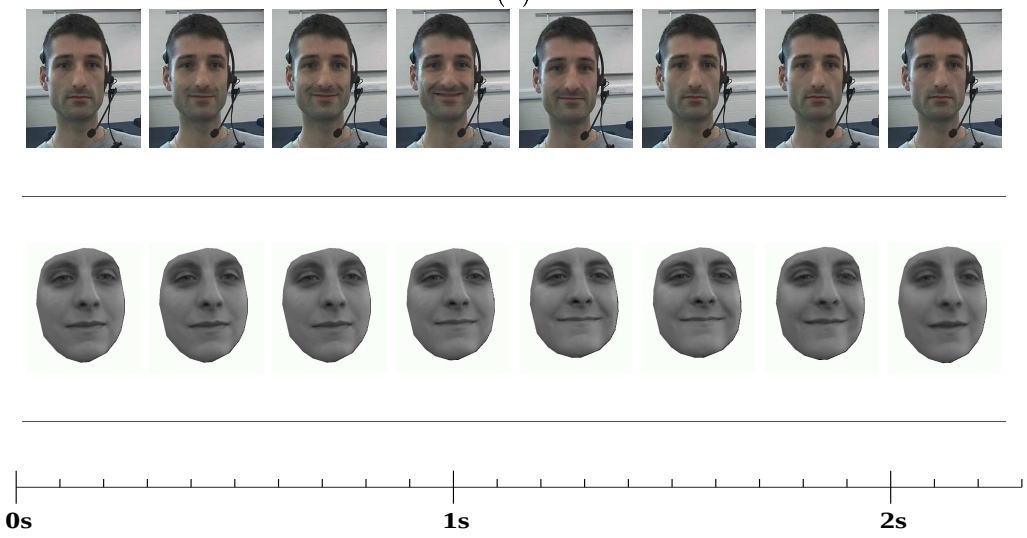
5.3 Application II: the listening head

For this application we are aiming for more than just a few greetings. We want the first speaker to talk freely and get a simple answer as a smile, a nod or any other simple acknowledgment based on the interactional quality and facial expression of the speaker.

The model is trained with sequences of two people. One talking and the other one listening. The acquisition was staged in the same way as the previous application. The first speaker's video and audio are processed while only the video is processed for the listener (see Figure 5.6). The speech model for the audio of the first speaker has been generalised to a one-dimension energy signal (see Subsection 3.2.1). After processing, the sequences are manually segmented into interaction clips. Each clip contains the first speaker's 'speech' and finishes with the second speaker's reaction.



(a)



(b)

Figure 5.4: In (a) the face as input saying "How do you do?" with the associated speech waveform and the synthesised face with the associated speech waveform. In (b) the face smiling as input with the associated speech waveform and the synthesised face with the associated speech waveform.

5.3.1 Experiment

In this experiment the system was trained with 20 interaction clips.

Those clips were extracted from a 3 minutes conversation where only speaker one was

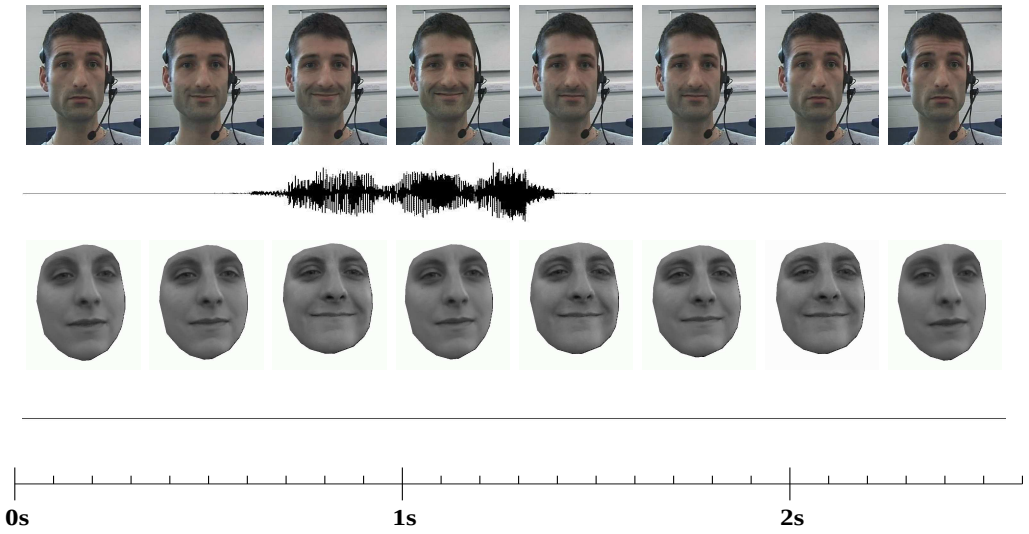


Figure 5.5: On first row the face smiling as input with the unusual associated speech waveform "how do you do" reversed and on second row the synthesised face responding

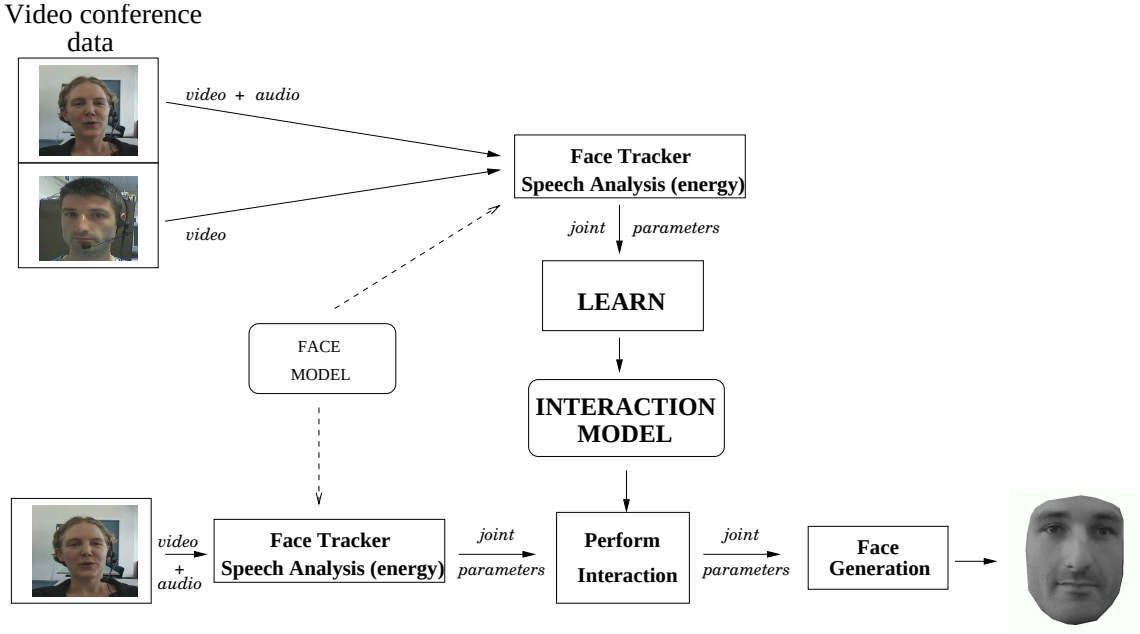


Figure 5.6: Listening Head learning system and application

talking and speaker two was listening. Each interaction clip lasts between 3.5 and 9.7 seconds.

The face model of the first speaker had 11 components (see Section 3.1) and the speech

had one component measuring the energy (see Section 3.2). The face model of the second speaker had 8 components. The variation in the number of components for the face tracking is due to the fact that the second speaker has very few mouth movement as he's only nodding and smiling. The number of component for the first speaker could be much higher if we were trying to get the precise movement of the mouth for all the complex speech but it has been reduced to get a general head shifting and a few facial expression with barely any lip movement. This is to make the tracking as general as possible. We don't want the tracking to be affected by what the speaker is saying, only by how he says it. The 20 interaction clips used represent a total of 1340 frames. The model was trained with 100 configuration prototypes and 400 behaviour prototypes. We end up with a Markov Chain of 353 connected states using 100 hypotheses for the propagation. The number of prototypes has been chosen to be very low to get a model as general as possible and able to deal with new utterances.

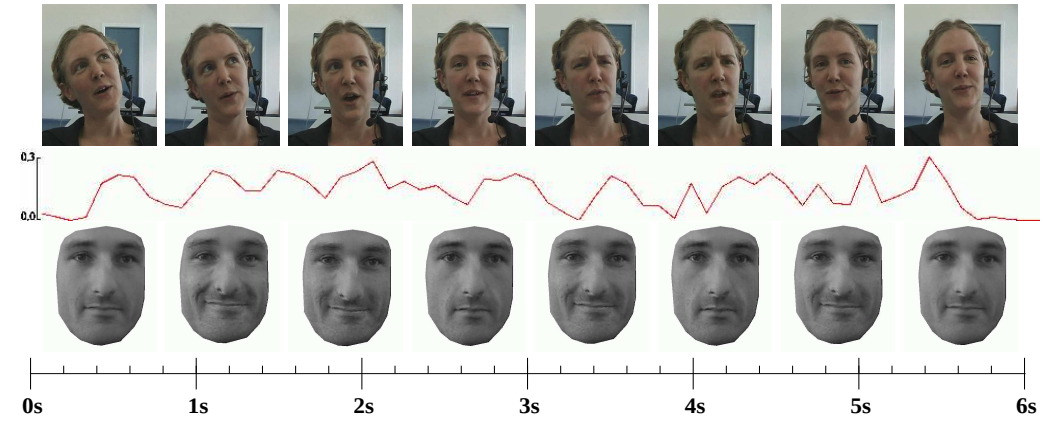
A slight variation in the algorithm is made. As the interaction clips used for training have large variation in their length, reaching an end state in the Markov Chain doesn't mean the interaction is over. The algorithm will loop from an end state to a new start state and keep going until the new input clip is over.

The model is tested with two new inputs of different lengths.

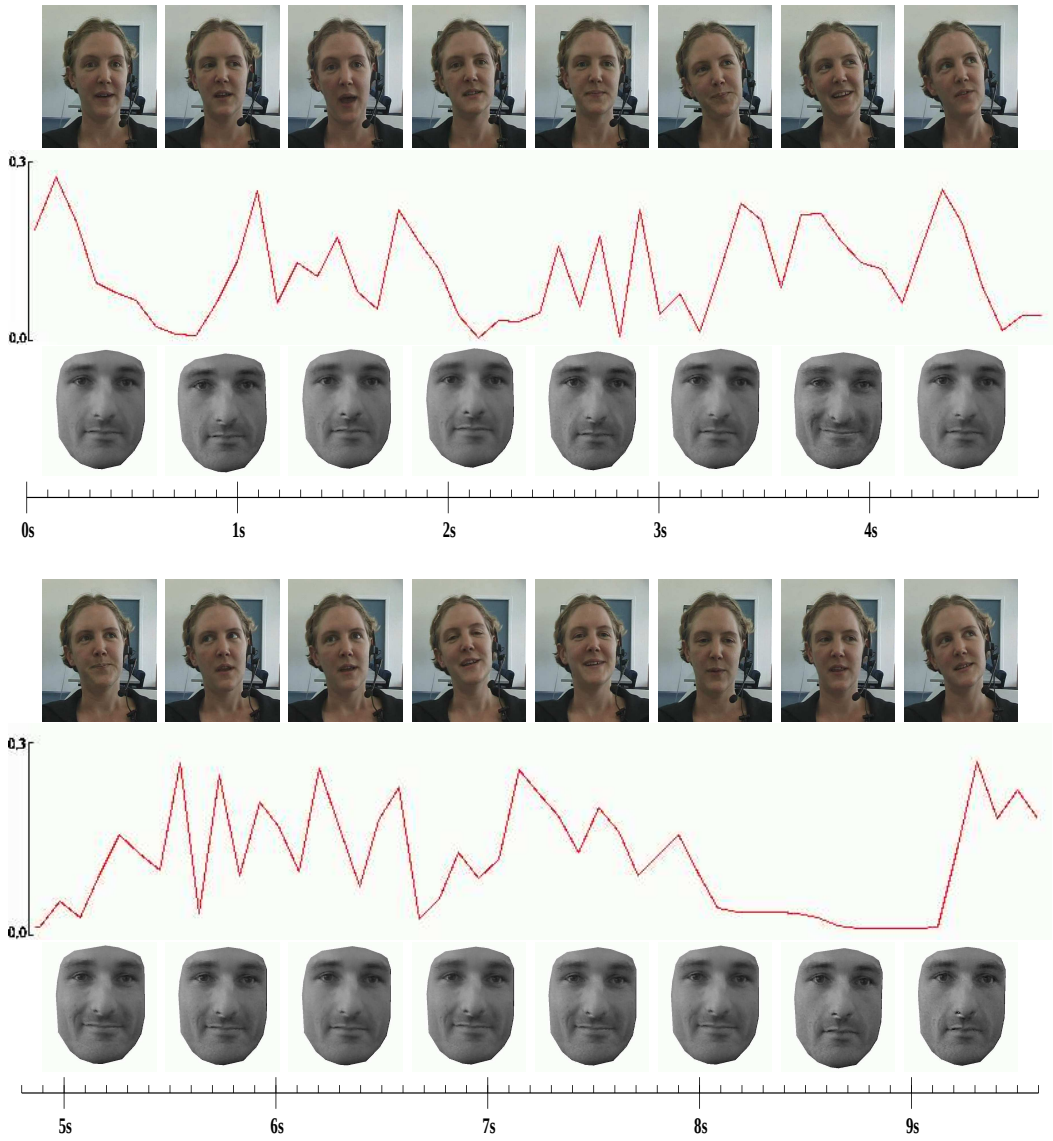
The experiment results illustrated in 5.7 show some different outputs. In the first result 5.7(a) the input was 6 seconds. The listening head reacts well to the talker.

In the second result 5.7(b) the input was 9.6 seconds. The algorithm had to loop 5 times to give the response. In both case the listening head reacted actively to the speaker's speech. Driven by the energy of the speech signal and the facial expressions it gives the illusion of an interested person.

Unfortunately, as it can be seen in Figure 5.8, with an energy signal forced to zero the model can still generate a convincing nodding and smiling head. This means that the validity of using facial expression/energy has not been proven. The listening head experiment is inconclusive.



(a)



(b)

Figure 5.7: (a) and (b) : At the top the speaker talking freely as input with the associated energy signal and at the bottom the synthesised face responding

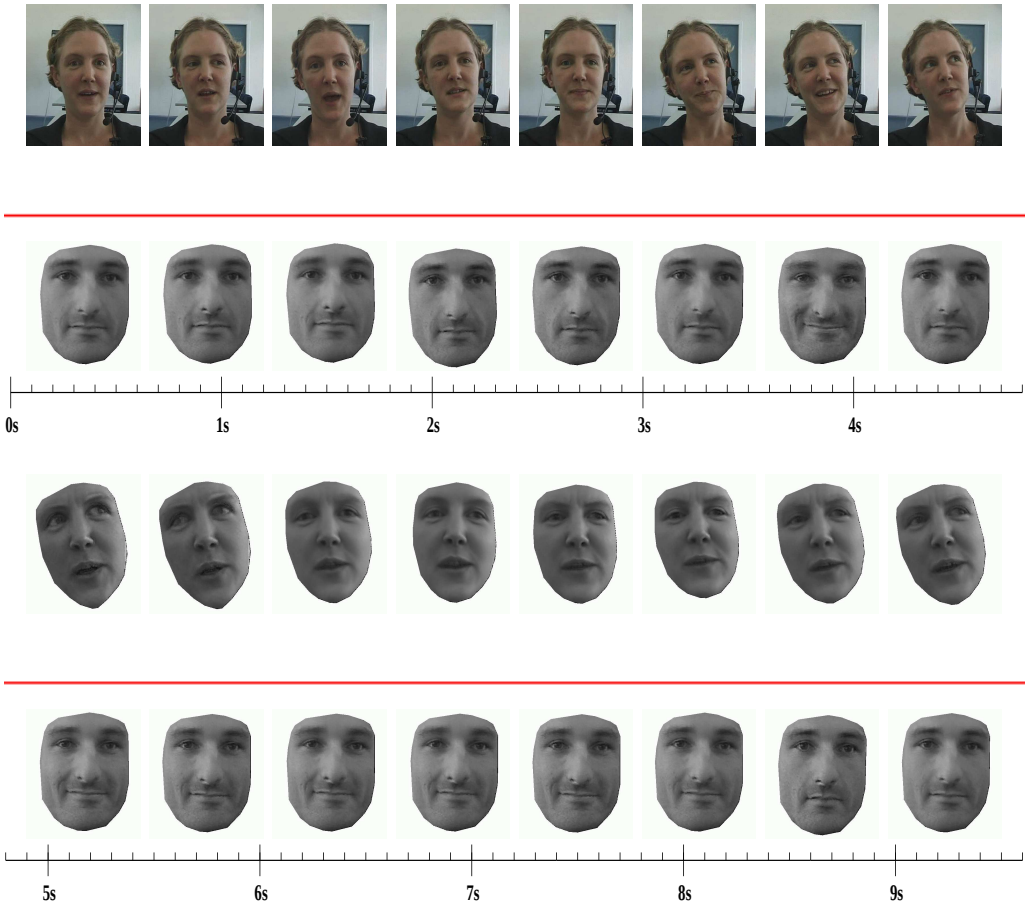


Figure 5.8: At the top the speaker talking freely as input with the associated energy signal and at the bottom the synthesised face responding

5.4 Application III: Assisting a partially sighted listener

In this application we generate, as output, a voice describing what the input face is doing. It is easy to see the various applications for wearable computers with video cameras for partially-sighted people and also video conferencing. People will be able to hear and be aware of what's happening in video conferencing events from a basic telephone.

We encode the joint behaviour of someone speaking (person 1) and a second person verbalising the facial expressions and the meaning of head gestures (person 2). This is achieved using a combined model of facial appearance (from person 1) and auditory waveform (from person 2). The model encodes non-verbal behaviours like nodding or smiling and

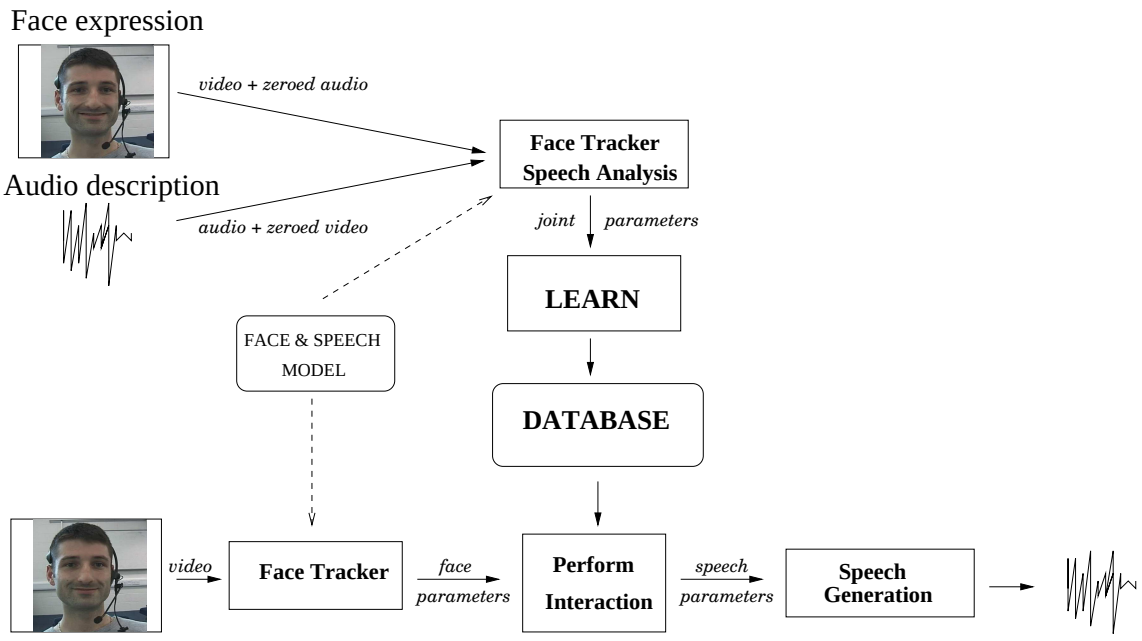


Figure 5.9: Facial expression descriptor learning system and application.

the sound model encodes a vocal description of these behaviours - the utterance 'yes' when the person is nodding, 'no' when they shaking their head and 'oh' when looking surprised, and 'smile' when they are smiling.

Person one is obtained from the visual components with a zeroed audio channel and person two from the audio components with a zeroed visual channel. - in practice we use the same person verbalising a recording of their own facial expressions.

In summary, the aim is to create a model of usual silent face expressions and head gestures associated with their vocal description (see Figure 5.9).

5.4.1 Experiments

In our experiments, we use 38 training video clips of non-verbal behaviours commentated as described. The utterances used are shown in Table 5.3. As the application will be used in conjunction with a normal conversation, it seems more natural, as explained before, to use the same person's voice joint to his or her facial expressions. Thus, the model created

Learning video clips		
Face expression	Vocal description	Number
Smiling faces	"Smile !"	9
Nodding faces	"Yes !"	9
Shaking faces	"No !"	9
Surprised faces	"Oh !"	10
-	-	1
	TOTAL	38

Table 5.3: Learning video clips : different face expressions

is the person's description of their own behaviour. Figure 5.10 shows the different kinds of expressions used.

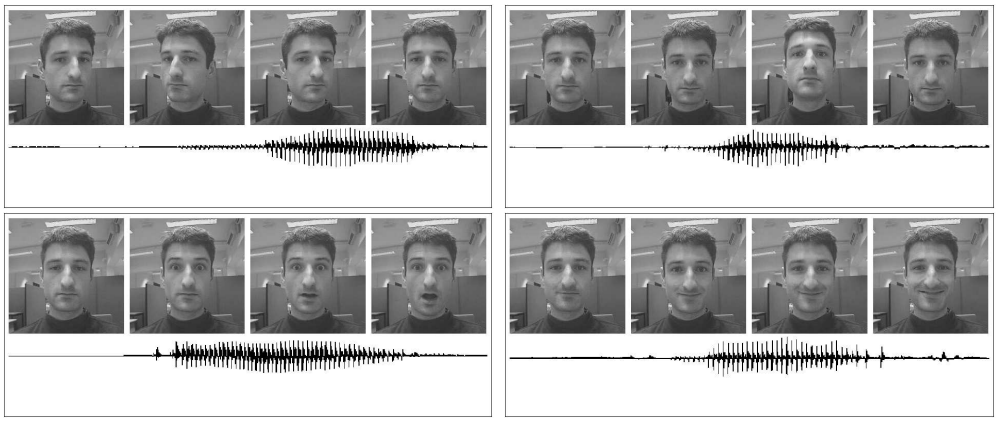


Figure 5.10: Four different expressions commentated and trained. Top left, a head shake associated with the sound signal "No". Top right, a head nod associated with the sound signal "Yes". Bottom left, a surprise face associated with the sound signal "Oh!". Bottom right, a smiling face associated with the sound signal "Smile"

Each interaction video clip lasts between 2.5 and 5 seconds.

The face of the speaker was tracked with a model based on 12 components (see Section 3.1) and the speech with a model based on 70 components (see Section 3.2). The 38 clips used represent a total of 1688 frames. The model was trained with 800 configuration prototypes

and 800 behaviour prototypes. We end up with a Markov Chain of 420 connected states using 100 hypotheses for the propagation.

The model is tested with two new inputs, a smile and a surprise face.

5.4.2 Results

Figure 5.11 and 5.12 show the results obtained with those two basic and frequently used facial expressions. Those expressions are visually understandable for sighted people and now, with the synthesised output, for partially-sighted people.

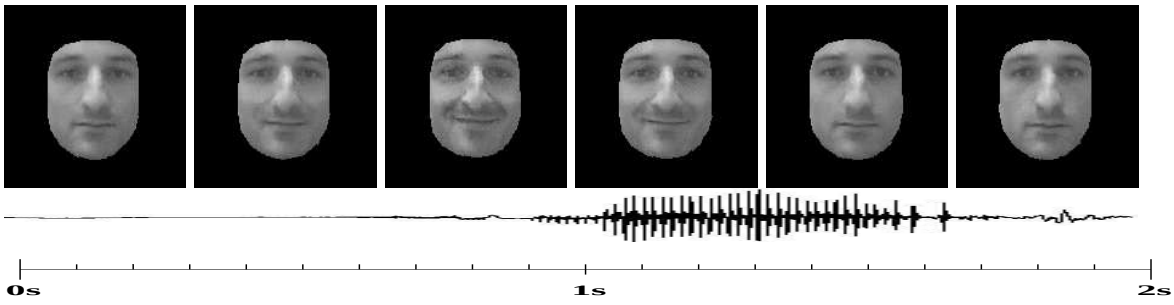


Figure 5.11: The face expression as input and the audio output description 'Smile'

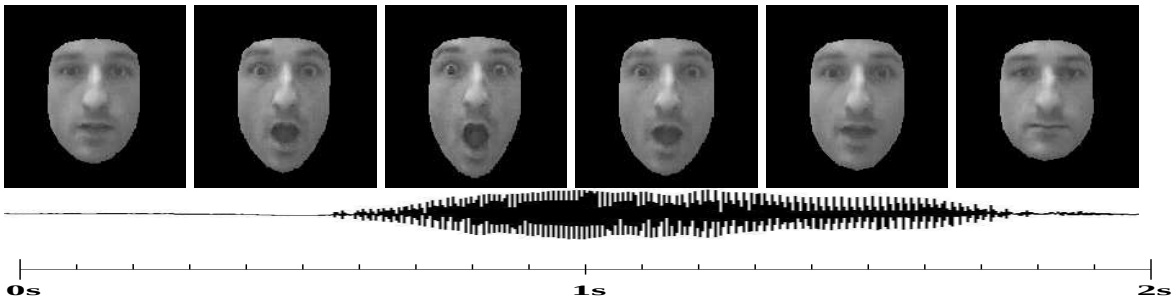


Figure 5.12: The face expression as input and the audio output description 'Oh !!'

As explained with previous application the timing is very important in this application and, as opposed to expression recognition software with synthesised speech, the output comments have a very natural response. We can observe in Figure 5.13 the kind of response timing we obtain. The vocal description output starts before the end of the smile.

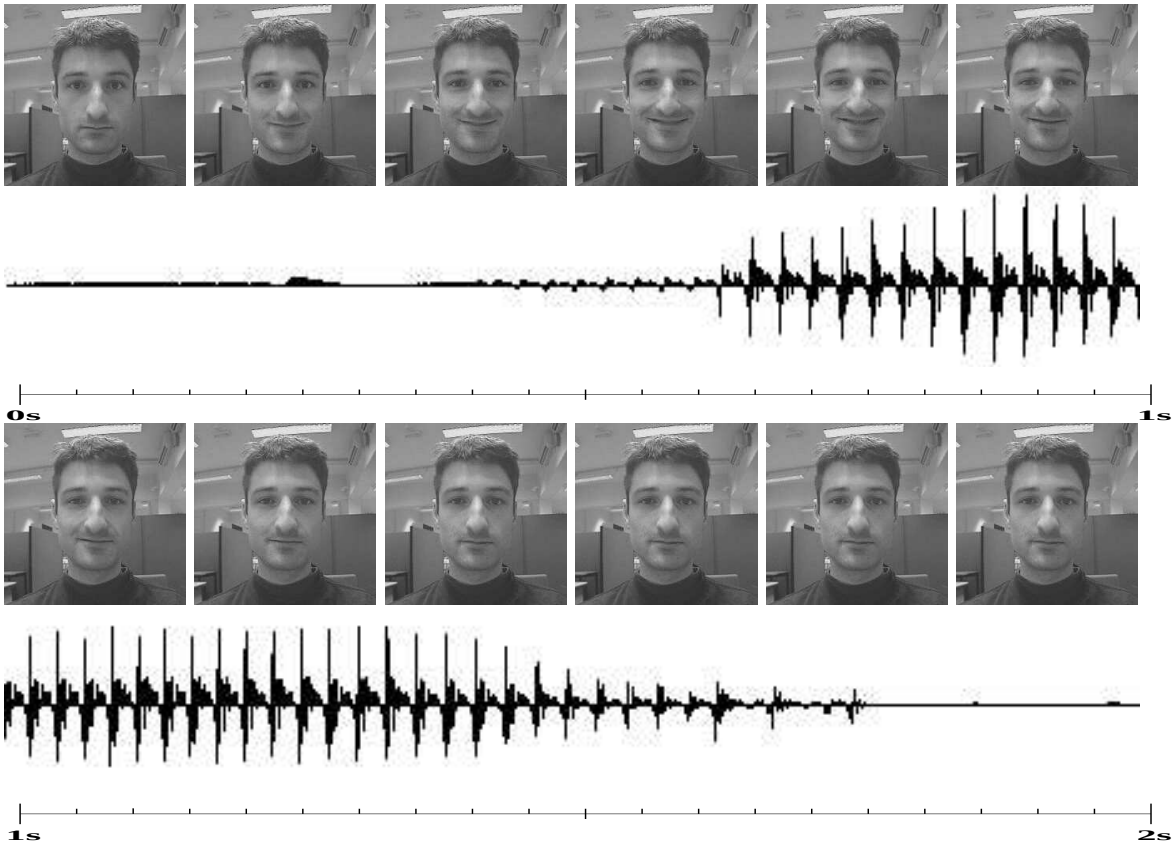


Figure 5.13: Timing for the vocal description response

5.5 Application IV : a behaviour filter

In this application we demonstrate the use of interaction models as behaviour filters. Any 'improper' behaviour can be detected and the application can take over and replace what is observed by what is 'proper'. By 'improper' we mean not 'observed' response.

In the talking Head and listening Head applications the models describe all the behaviour seen in the training interaction clips. If a new input clip cannot be processed correctly by the model this means it's an incorrect one and can be erased or altered.

5.5.1 Method

With a large and varied training data set all the basic *action/reaction* pairing should be modelled. Any interaction clip of a *correct* behaviour should be able to go through the Markov Chain keeping the error $E(F_t, T_t)$, where T_t is speaker one AND speaker two's joint observation, under a fixed threshold.

This *behaviour threshold* (BT) can be used as a filter for a correct behaviour. Any incorrect behaviour would be detected when the error $E(F_t, T_t)$ crosses the threshold.

To avoid filtering proper behaviour when noise occurs in the tracking, we filter on a smoothed accumulated error E^s :

$$E_t^s = \lambda E(F_t, T_t) + (1 - \lambda) E_{t-1}^s \tag{5.4}$$

with $E_0^s = E(F_0, T_0)$ and $\lambda = 0.2$

The algorithm to go through the Markov Chain is :

1. Generate a set $\mathcal{X}_0$ of N hypotheses to represent the initial prior, where $\mathcal{X}_0$ is obtained under sampling with replacement from the initial state distribution π.

2. For each $F \in \mathcal{X}_t$, use the error $E(F_t, T_t)$ to calculate the likelihood of the hypothesis using Equation $P(T_t|F_t) = exp\left(-\frac{E(F_t,T_t)^2}{2\sigma^2}\right)$.

3. Use relative likelihood values to weight sampling from $\mathcal{X}_t$, the prior, resulting in a set $\mathcal{Y}_t$ of N hypotheses representing the posterior distribution.

4. Update E^s with the error $E(F_t, T_t)$ from the hypothesis in $\mathcal{Y}_t$ with maximum likelihood. If $E^s \geq BT$ then signal improper behaviour.

5. Generate a new set $\mathcal{X}_{t+1}$ of N hypotheses to represent the new prior, where each $F' \in \mathcal{X}_{t+1}$ is a stochastic extrapolation at time $t+1$ from $F \in \mathcal{Y}_t$ using the transition probabilities $\mathcal{T}$

new input clip		
speaker one	speaker two	length
"How do you do ?!"	"Fine !"	2.4s
"How do you do ?!"	"Yawning..."	2.4s
"How do you do?... Hi!?"	"Fine !"	2.4s

Table 5.4: tested interaction clip for the behaviour filter

6. Repeat steps 2-5 until the interaction is complete.

5.5.2 Experiment

To test the filtering we use the interaction model built for the second experiment of application 1, see Section 5.2. The model was trained to recognise and generate simple greetings (see Table 5.2).

We input three different new inputs of the same length (see Table 5.4). The first input is a common interaction that the model should recognise and process without much difficulty. The two others, on the other hand, are rude behaviours. The response by the second speaker who's not paying attention in the second clip and in the third clip by the first speaker who's not listening to the answer and greets again.

In Figure 5.14 we can observe the accumulation error through time of the input clip. If we fix the Behaviour Threshold at 2.3 all the interaction clips encoding an 'improper' behaviour would cross it and be detected.

The filter is actually an 'improper behaviour' detector. When the threshold is reached the application can inform that the interaction clip doesn't fit the model. From that point there are many possibilities :

- The application can take over and generate the appropriate virtual talking head by using T_t^B

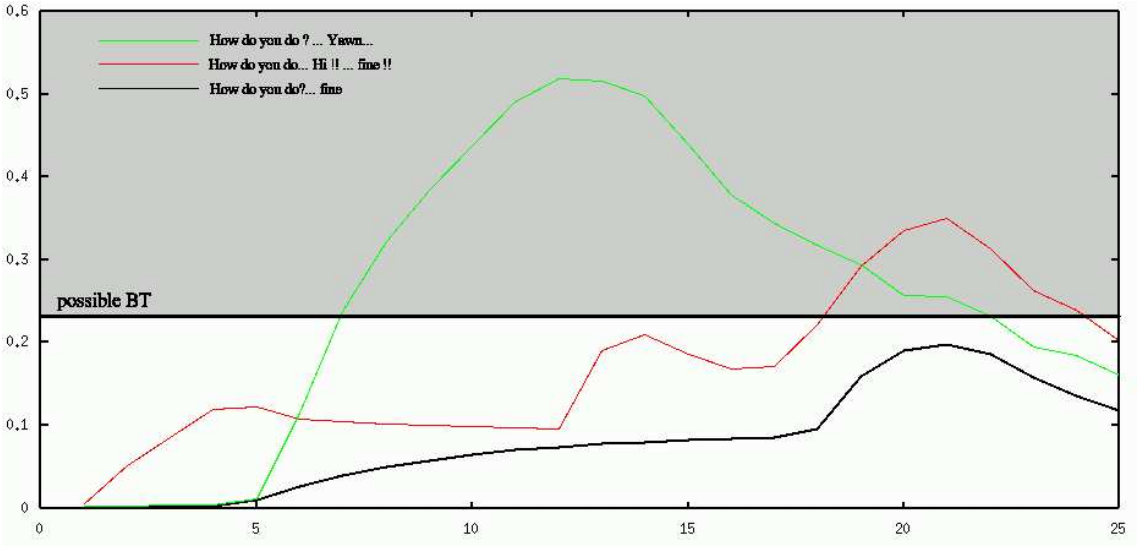


Figure 5.14: E^s values (vertical axis) for the three different inputs over time (unit is in frame)

- If the original interaction clip was nearly finished when the inappropriate behaviour was detected, the application can stop and re-input the clip from the beginning with only F^A visible and generate a proper response like application 1.

- The threshold crossing can be seen as a warning and the user, if the interaction clip was acquired in real time, can be informed that they are doing wrong. Or the clip can be rejected if the application is working off line.

5.6 Evaluation of the Interaction quality

In application 1, we generate a synthetic output claimed, in previous section, as realistic. To test this statement, an evaluation of the interaction quality of the application has been carried out.

The method used was a user evaluation marking. 20 subjects had to judge the interaction quality of interaction clips generated by application I Figure 5.15.

Figure 5.15: User evaluating the interaction

5.6.1 Setup

Synthetic Interaction clips were generated from the two different models described in Section 5.2. 'Synthetic' meaning generated responses from real observations. Real interaction clips were taken from the two different training datas. 'Real' meaning observation, after face tracked and sound analysed, but without any further processing except graphical reconstruction.

As both types of clip were going to be mixed for the evaluation the best synthetic clips generated were chosen to join the real ones in the 16 interaction clips to evaluate (see Table 5.5)

These 16 interaction clips were duplicated 6 times and randomly arranged to form a 7 minute video, so each subject will have to mark the same clip several times. 20 subjects were selected and spent 10 minutes each doing the evaluation.

The subjects were told to watch 1 minute of the movie first without marking to get used to the kind of interaction they were going to mark. They were also told that half of them

evaluation clips		
greeting	real/synthetic	Number
"Hello ?!"..."Hello !"	real	2
"Hello ?!"..."Hello !"	synthetic	2
"How do you do ?"..."Fine !"	real	2
"How do you do ?"..."Fine !"	synthetic	2
"Hi ?!"..."Hi !"	real	2
"Hi ?!"..."Hi !"	synthetic	2
Smile...Smile	real	2
Smile...Smile	synthetic	2
	TOTAL	16

Table 5.5: interaction clips used for the evaluation

are real observations of two people interacting.

Before starting the marking the subject instructions were :

- Do not point out which ones are synthetic and which one are real but mark all of them in interaction quality.

- The marking scheme is in 7 points. 0 for neutral, -3 when it looks very unrealistic and +3 when it looks very realistic.

- Pause the movie if thinking is needed and even go back if the interaction has to be seen again.

5.6.2 Results

We can observe in Figure 5.16 the distribution of the marks for the real and synthetic interaction clips. The mean score for both real and synthetic interaction clips errs on the side of realistic. The fact that the real interactions have been marked higher than the synthetic ones was expected. Nevertheless, the average marking difference between the

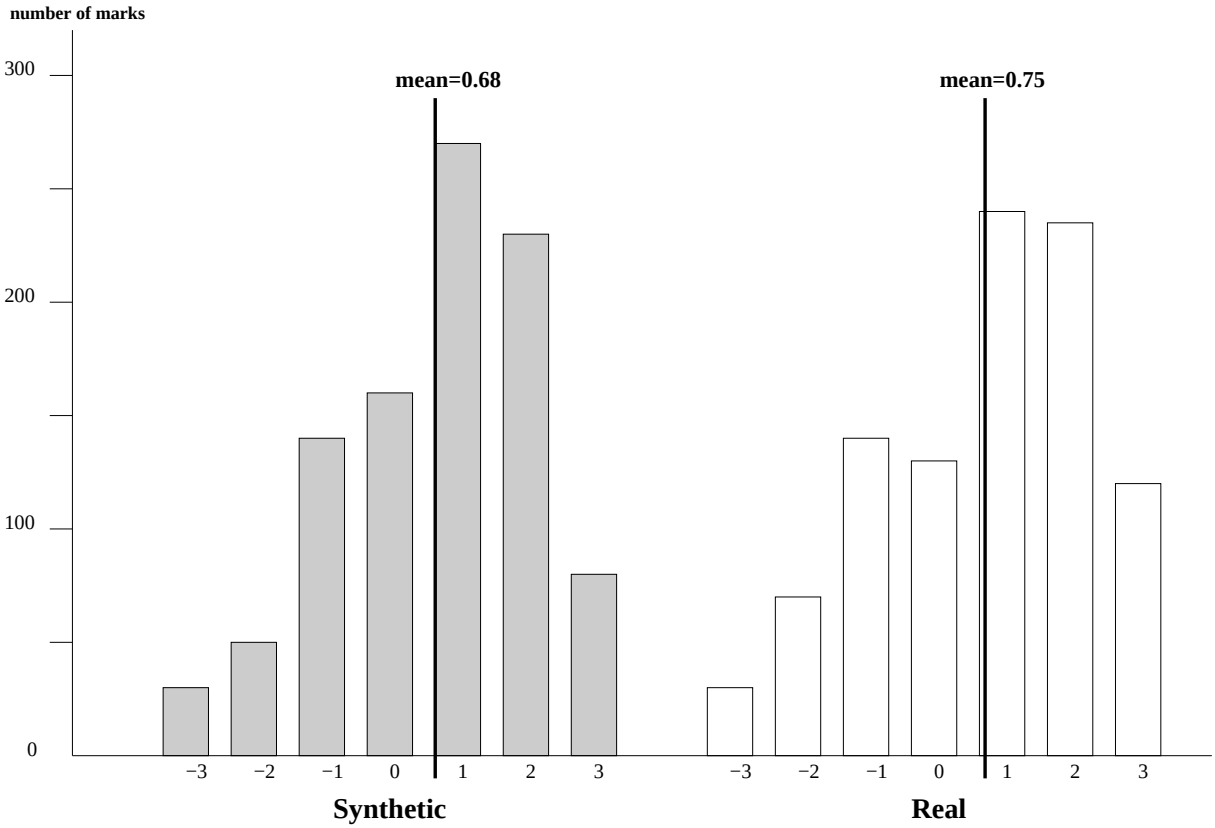


Figure 5.16: Distribution of the marks

real and synthetic interactions is low enough so we can say that the synthetic clips are admited as realistic as the real ones.

The marks '-3' and '-2' (looking unrealistic) is, in both real and synthetic interactions, occurring with lowest frequency. The mark used the most was '1' which can be explained by the fact that two faces without hair staring at the subject and greeting each other is difficult to mark as 'very realistic'.

If we observe the variation graph in Figure 5.17 we can notice that the real interactions, on average, caused more trouble to the subjects, especially the greeting 'How do you do'.

In the case of the synthetic 'hello' variation, the high value can be explained. One of the two synthetic 'hello' had the specific timing seen in the third experiment of Subsection 5.2.1, which we called 'the fast answer'. The fact that the response overlapped the question might have troubled some of the subjects and created that big variation in the marking.

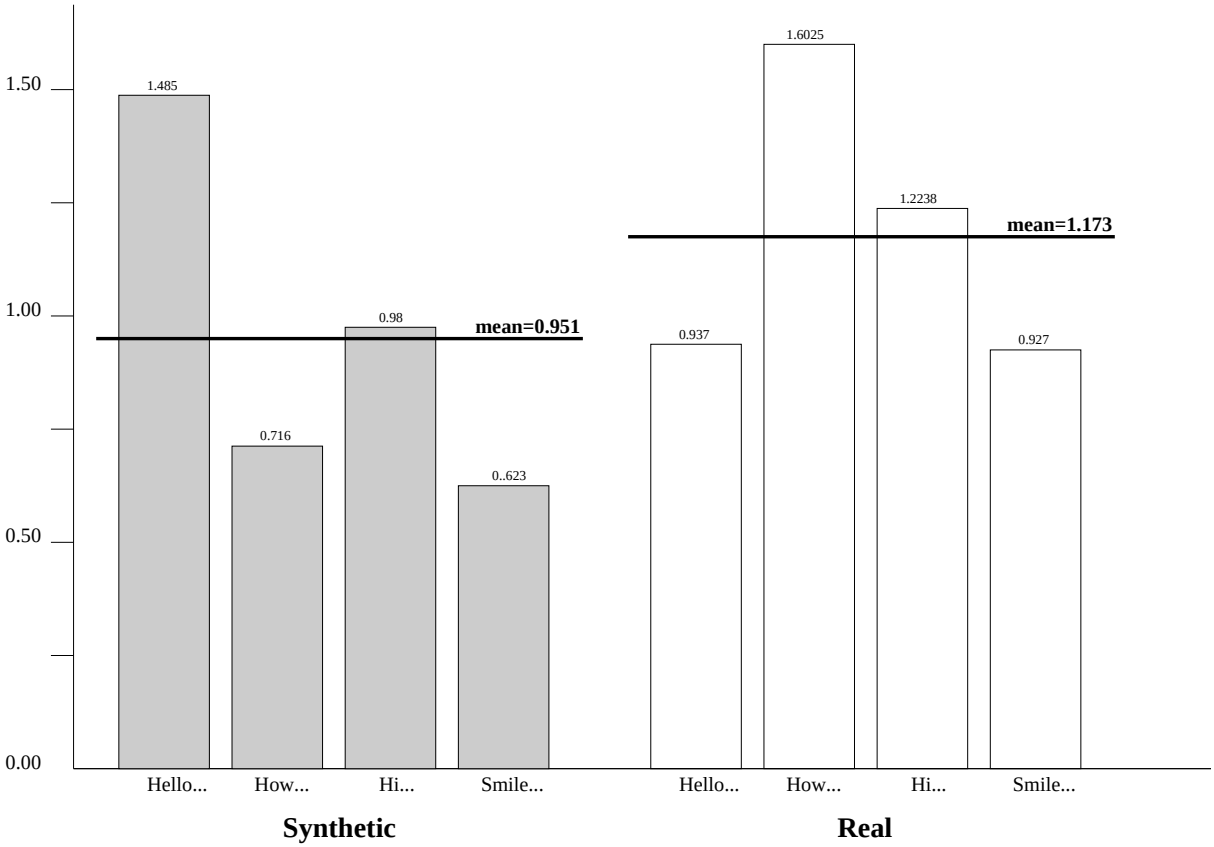


Figure 5.17: Variance of the marking for each type of greeting

The complete results of the evaluation can be seen in appendix A

The use of the behaviour model to generate a talking head has been demonstrated. Various possible applications have been presented, performed and evaluated. In the next chapter the future of this research will be discussed.

Chapter 6

Conclusion

Our aim was to create a virtual talking head which learns from observation of real human conversations. The results shown in Chapter 5 validate our method. The overall framework linking speech with video within a reactive system has been demonstrated. The synthetic talking head, application I, responds appropriately and with correct timing to simple greetings with variations in facial expression and intonation. Other applications of the behaviour model have been presented: application II, the listening head, Application III, the partially sighted people assistant and application IV, the behaviour filter. The results show some encouraging results even if in the case of the listening head the experiments were inconclusive. The behaviour filter has been introduced, using the behaviour model database to signal any incorrect interaction sequence and some experiments show possible applications.

6.1 Discussion

We have not demonstrated the limits of application I and III. The full encoding of speech vocabulary must have a limitation in our applications. We don't know how high we can increase the number of greetings in application I and the number of expression descriptions in application III before the method fails to generate responses. The number of training sequences will have to increase and so will the number of prototypes needed to encode

them. The number of ramifications in the Markov Chain will get so high that the propagation won't be able to generate sensible responses. If we keep the number of prototypes low the quality of the response generation is going to drop until unrecognisable answers will be obtained.

A correlation has been shown in Chapter 5 between the length of the greetings. This correlation was observed in the training data. We showed that synthesised answers kept that correlation true to prove the quality of the model. What if some other interaction patterns were observed in the model. It would be an interesting way to extract, prove or discover some human interaction behaviour pattern by automatically generating an interactive model from human interaction databases and synthesising some interactions.

Application II is probably the most promising aspect of this research. The generalisation of the interaction is the most interesting part. Even if the method doesn't offer a full intelligent conversation with the machine the illusion is good enough to encourage the user to interact in the most natural way. This application needs more training sequences (see below) while the number of prototypes has to stay low. Also, the interaction clip size is an important matter that hasn't been dealt with yet. It is difficult to decide, in the case of a speaker talking freely and listener nodding, when the interactions start and when they end. This leads to the supervisor proposed in the next section.

If we want to test the limit of application I and III and to experiment more on application II we need to sort out the training data set size problem. To get a large amount of interaction data we need to be able to collect them from different people. We need a more general model including interaction between lots of different individuals. A solution would be to build a more general face tracker and speech analyser able to deal with different individuals. But the precision in the lips and facial expression has to stay the same. It seems difficult to build such a general model and keep that kind of tracking precision. Another way would be to keep individual models and map them into a single one which I will use to build the interaction model. Multiple individuals could then contribute to the final global interaction model.

6.2 Work extension

So far the face tracker uses and generates grey scale faces. A color model would probably not increase the performance of the tracker but would generate a more realistic output face. Furthermore, some subjects of the evaluation (see Section 5.6) pointed out that two floating faces talking to each other do not look particularly realistic. The use of a fixed or modelled background containing the rest of the face, hair and neck, on which the generated face is overwriten would make it look more realistic.

One aspect of the work done in application IV could be exploited in a different way. The joint model of speech and facial expression could be used as a biometric system. We model a person (face + speech) from a series of simple sentences with different timing and intonation. Then only that person would be able to input their face and voice, pronouncing the valid sentence, in the system and pass the validation test. A sentence would be validated when the Markov chain has been traversed until reaching an end state without the smoothed error E_s crossing a fixed threshold. Only a person with the same face, knowing the sentences and able to imitate the voice, intonation, timing and facial expression can pass the test.

Applications I and II need more thorough evaluation, as explained in the previous section. A simple aspect of the face tracker hasn't been fully used, the scale S. So far, this affine transformation was used to help the tracker initialising. None of the interaction sequences used in a training data set has large scale variations. The scale, which can represent the proximity of the face, can track and reproduce faces getting closer or further back. This kind of behaviour expresses emotions and also is used to grab the attention of a speaker (see Figure 6.1).

The only way, so far, to have a general conversation with one of our talking heads is to use application II. Unfortunately, the response from the machine is only an 'acknowledging' head. If we want to keep the property of application I, full speech answers, or maybe mix application I and II into a single one, we need a supervisor. Something above all the methods which would switch between models and application and help the conversation to go forward. This supervisor would need a speech interpreter and maybe a manually set

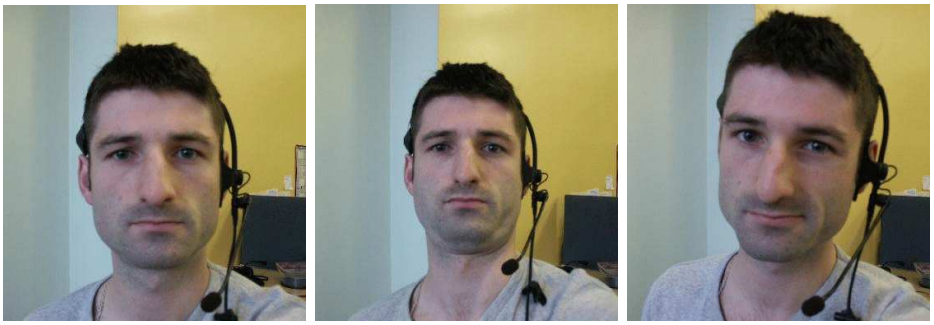


Figure 6.1: The proximity of the face can express emotions but also be used to grab the attention

behaviour rule-base. This would not be trying to step back to a type III talking head (see section 2.1.3) but adding a rule-based system on top of an automatically learnt Type IV talking head.

Appendix A

Evaluation tables

- Evaluation raw results
- Evaluation synthetic results
- Evaluation real results

	subject 1	subject 2	subject 3	subject 4	subject 5	subject 6	subject 7	subject 8	subject 9	subject 10	subject 11	subject 12	subject 13	subject 14	subject 15	subject 16	subject 17	subject 18	subject 19	subject 20	SUM/20
real hello 1	0	1	1	1	2	2	1	2	1	2	1	1	2	3	1	2	3	0	0	1	2.5
synthetic smile 1	-1	0	-1	-1	0	1	1	1	-3	0	2	1	1	1	-1	2	2	1	1	2	0.9
real hello 2	1	1	2	1	1	3	2	-1	2	2	-1	-1	0	2	0	1	2	-1	1	-1	0.4
synthetic hello 2	-2	1	0	-1	2	2	0	1	2	-3	2	-1	-1	-2	2	2	1	1	-1	1	0.6
real smile 1	0	-1	-3	-1	0	-1	1	2	-2	-1	-2	0	2	0	2	1	-1	1	-2	1	-0.4
synthetic hello 1	2	2	1	0	0	3	0	1	-3	-1	2	-2	0	3	2	0	-2	2	1	-2	0.9
real smile 2	1	2	-2	2	1	2	3	2	2	-1	1	0	1	3	0	1	2	1	0	1	2.2
synthetic smile 2	0	0	-2	2	2	2	2	3	2	2	0	0	2	2	1	2	1	2	1	2	2.6
real hi 1	2	3	3	1	3	3	3	3	3	1	2	1	2	2	3	3	-3	3	2	3	4.3
synthetic how 1	3	2	2	0	-1	2	-1	3	2	2	0	-1	2	1	3	0	1	0	2	-2	2
real hi 2	0	2	3	1	1	0	0	3	2	1	2	0	1	0	2	2	-1	2	1	3	2.5
synthetic hi 2	-2	1	-1	1	1	1	2	3	-2	2	3	0	2	3	3	1	0	2	2	1	2.3
real how 1	-2	0	-2	1	2	-2	2	2	2	2	0	-1	0	1	3	0	-1	1	3	-3	0.4
synthetic hi 1	0	-1	1	0	-1	2	-1	1	1	1	-1	-1	1	2	2	-1	-3	0	1	-2	0.1
real how 2	0	0	-3	1	-1	1	1	2	1	-1	-1	-2	3	1	-2	-2	0	-1	2	-1	-0.2
synthetic how 2	1	2	-3	1	2	2	1	2	0	-1	2	0	1	0	-1	-2	0	1	1	1	1
synthetic how 1	-1	1	0	0	-1	3	-2	2	-1	-1	-1	-1	1	0	-1	-1	1	0	0	1	-0.1
real smile 2	0	-3	1	1	0	3	3	3	1	2	2	1	1	3	1	0	3	2	-1	2	2.1
real hello 2	-2	1	2	0	2	3	1	-1	-3	0	3	-1	0	3	1	-1	-1	1	-1	2	0.9
real hello 1	-1	3	3	1	2	3	-1	2	2	2	3	-1	1	2	-1	1	1	2	1	3	2.8
synthetic smile 1	2	-1	3	2	2	3	2	3	0	1	1	1	1	2	-1	1	2	2	0	1	2.7
real how 2	0	1	-3	0	-1	2	1	3	0	0	0	-2	2	2	-2	-2	0	-1	1	-3	-0.2
synthetic hello 1	1	3	1	1	3	2	1	-2	-3	1	2	-2	-1	2	3	2	-3	2	2	2	1.7
real hi 2	2	3	2	2	3	0	3	3	3	2	2	2	1	3	3	3	-2	3	2	1	4.1
real how 1	0	-1	0	1	-3	-1	2	1	2	2	-2	-1	1	1	2	-1	1	1	1	0	0.6
real hello 1	-1	1	1	0	2	2	1	2	2	3	3	1	3	3	-1	1	1	2	2	2	3
synthetic hi 1	-2	1	-1	1	0	2	2	1	3	3	-1	-1	1	0	1	1	-3	1	1	0	1
synthetic hello 2	-3	0	1	-1	1	3	1	-1	1	2	0	-1	-1	-2	1	0	1	0	-3	0	-0.1
synthetic smile 2	0	2	0	2	2	-1	0	2	0	3	0	1	-1	-1	2	1	3	2	2	1	2
real hello 2	0	2	1	1	1	3	-1	-2	-1	2	-1	-2	-1	1	0	2	0	0	0	0	0.5
synthetic how 1	-1	1	1	2	-1	-1	-2	1	2	3	-1	-1	1	0	1	0	1	0	0	-3	0.3
real hello 1	-2	2	0	1	2	3	1	2	1	3	2	-1	1	2	1	1	1	3	2	2	2.7
synthetic hi 2	-1	2	3	2	2	1	1	2	0	3	3	1	2	2	3	1	0	2	2	3	3.4
synthetic how 2	2	2	3	3	2	2	-1	3	2	2	2	2	2	0	3	0	1	1	3	3	3.7
synthetic smile 1	0	-1	3	1	2	1	1	2	1	1	2	1	0	2	2	1	1	0	1	0	2.1
real hi 2	2	2	2	1	2	3	3	2	2	-1	3	2	-1	1	3	3	-3	3	2	3	3.4
real hello 2	1	2	1	-1	1	2	2	-2	-2	2	0	-1	0	0	2	1	1	1	1	0	1.1
synthetic how 2	0	1	-1	1	2	0	1	3	2	2	2	1	2	1	2	0	2	2	1	2	2.6
real hi 1	0	2	2	1	2	2	1	3	3	2	2	2	1	3	3	-3	3	3	0	2	3.3
real how 2	-1	2	-3	2	-1	1	1	2	-1	-1	-2	0	1	1	-1	-2	2	0	2	-3	-0.1
real smile 1	-1	-2	-3	1	1	1	-1	-3	0	-2	1	0	-1	-2	-2	1	2	1	1	0	-0.8
synthetic hi 1	2	1	-1	0	-2	0	-1	-2	1	-1	1	-1	-1	1	1	1	0	2	0	1	0.2
real how 1	3	-1	-1	1	-2	-1	0	-1	-1	-1	-1	-1	1	0	-1	0	1	1	0	-2	-0.6
real smile 2	0	-2	-2	2	1	2	2	2	2	-1	1	1	0	3	2	1	3	2	0	2	2.1
synthetic hello 1	2	2	2	-2	1	3	1	-3	-3	-2	3	-1	-1	1	-1	1	-3	1	2	2	0.5
synthetic how 2	1	2	-2	2	1	2	2	3	3	-1	2	1	1	1	2	1	2	1	1	3	2.8
synthetic smile 2	0	1	-1	1	0	1	2	3	2	1	0	1	2	1	1	1	3	2	0	1	2.2
real smile 1	0	-2	-1	0	2	-1	-1	-1	-1	1	-1	-1	-1	-1	-2	1	1	2	1	0	-0.5
synthetic how 2	1	2	-1	1	2	1	1	2	3	-1	2	0	1	1	2	0	2	1	2	1	2.3
synthetic hi 2	0	1	-2	0	2	1	1	1	-2	0	1	0	1	2	3	3	1	2	2	2	1.9
synthetic smile 1	0	1	2	1	1	0	1	-1	-1	-1	1	0	0	2	-3	1	1	1	1	0	0.7
real hi 2	1	0	3	1	2	2	1	3	-2	1	2	0	2	0	2	3	-1	3	0	2	2.5
real hello 2	-1	2	1	-1	1	2	0	-3	0	0	-1	-2	1	2	2	0	2	0	3	0	0.8
real how 1	2	1	-1	1	-1	0	1	1	-1	1	-2	-2	2	1	2	0	1	2	2	-3	0.7
real hi 1	2	2	1	1	2	3	2	2	1	-2	2	1	2	2	3	3	-3	2	3	1	3
real how 2	1	2	-3	1	-1	1	-1	3	0	0	-1	-2	1	1	-1	-1	1	-1	1	-2	-0.1
real hello 1	1	2	2	2	2	3	0	2	3	2	2	-1	2	2	2	2	3	3	2	1	3.7
real hi 1	-1	2	0	1	2	2	2	3	0	1	3	1	1	1	3	3	-2	3	1	3	2.9
synthetic hello 2	-2	-2	0	-2	1	1	-1	-3	1	2	1	-1	1	-2	0	1	3	0	-1	2	-0.1
synthetic how 1	-1	1	-2	1	0	3	1	2	2	2	-1	-1	1	1	-1	-1	2	0	0	-1	0.8
synthetic hi 1	0	2	-2	1	-1	1	2	-3	3	0	-2	-2	-1	2	-1	1	-1	0	2	1	0.2
real smile 1	-2	2	0	-1	1	0	0	-2	-2	2	-1	1	0	1	-3	1	-1	2	1	0	-0.1
synthetic hi 2	0	2	-1	1	2	2	0	-1	-1	1	2	0	2	1	3	2	2	2	0	2	2.1
synthetic smile 2	2	2	-1	0	1	0	2	3	2	3	0	0	2	0	2	1	3	2	1	1	2.2
real smile 2	1	3	0	1	2	2	1	1	-1	-2	0	0	-1	3	3	1	2	2	-1	2	0.7
synthetic hello 2	-1	1	1	1	0	2	0	-3	-3	1	-2	-1	2	-2	1	0	2	1	-3	2	-0.1
real how 2	0	2	0	1	-1	1	0	3	0	-1	1	-2	2	1	1	0	2	-1	0	-1	0.8
synthetic hi 1	3	2	0	-1	-1	3	1	-1	0	-2	-1	-1	1	2	1	1	1	1	1	0	1
synthetic smile 2	2	2	1	1	2	3	3	3	2	-1	0	1	1	1	-1	2	3	2	-1	1	2.7
real hi 2	-1	2	2	1	2	2	1	3	-1	1	1	1	1	1	3	2	-1	3	0	2	2.5
synthetic hello 1	2	3	2	-1	2	3	2	-2	-3	-1	3	-1	-1	3	2	1	-3	3	0	1	1.5
real hi 1	1	3	2	1	2	1	2	3	2	2	3	1	2	2	3	3	-2	3	2	1	3.7
real smile 2	0	0	1	1	2	3	2	1	2	-1	1	1	-1	3	-3	2	2	3	1	-3	1.7
real how 1	-1	2	0	2	-1	-1	1	2	2	0	1	-1	2	2	1	0	1	2	-2	-3	0.9
synthetic hi 2	-1	2	0	1	2	1	1	2	0	-1	2	-1	1	2	3	1	2	2	2	-2	1.9
real smile 1	0	-1	1	-1	1	-2	-1	1	0	-1	-1	0	-2	2	-3	1	-1	1	0	-1	-0.7
synthetic how 1	1	1	-1	1	0	1	0	1	2	-1	0	-1	2	2	1	0	1	-1	1	-1	0.9
synthetic hello 2	-1	2	-1	-1	1	0	0	-1	-2	2	-1	-1	2	-2	0	-2	0	-3	2	0	-0.8
synthetic hello 1	-1	1	-2	-1	2	3	1	-1	-2	-1	3	-1	-1	0	2	1	-3	0	-1	0	-0.1
synthetic smile 1	0	1	-1	1	1	-1	2	1	1	0	1	-1	-2	1	-3	1	1	1	0	1	0.5
real hello 1	-2	2	0	-1	2	2	2	3	0	2	3	-1	3	2	2	1	2	3	1	1	2.7
synthetic smile 1	0	2	2	-1	2	0	2	3	2	1	1	0	2	2	-3	1	1	2	2	2	1.5
real hello 2	1	1	1	0	1	2	-1	-1	-1	-1	1	-1	1	0	2	1	2	1	1	0	1
synthetic hello 2	-1	0	-1	-1	0	-1	0	1	1	-1	-1	-2	0	-1	-1	1	1	1	-2	1	-0.6
real smile 1	2	2	3	-1	1	-3	-1	1	1	0	0	1	-2	1	-1	1	-1	1	1	1	0.7
synthetic hello 1	0	0	1	0	0	-2	0	0	-3	-3	-3	-2	-3	0	-3	-1	0	-3	-3	-2	-2.7
real smile 2	0	0	1	1	1	1	1	3	1	-3	-1	0	-2	1	-1	1	2	1	-2	0	0.5
synthetic smile 2	1	2	1	2	1	1	2	3	1	1	1	0	0	2	1	1	3	1	1	2	2.7
real hi 1	2	3	2	2	2	2	3	3	3	2	2	1	1	1	3	2	-2	3	0	3	3.8
synthetic how 1	1	2	2	1	0	3	2	3	0	-1	1	-1	1	2	2	1	2	0	1	2	2.4
real hi 2	0	2	2	1	2	3	2	3	3	2	3	1	1	1	3	3	-1	2	-1	2	3.4
synthetic hi 2	-1	2	1	1	1	3	1	2	0	3	3	1	2	1	3	1	2	2	1	3	3
real how 1	-1	0	-1	1	-1	2	1	1	1	-1	-1	-2	2	0	2	-1	2	1	2	-2	0.5
synthetic hi 1	0	1	-3	1	-1	2	1	-1	3	1	-3	-1	0	2	1	1	-1	1	2	2	0.8
real how 2	1	2	-2	0	0	1	2	3	0	-2	-1	-2	2	1	0	-2	2	-1	3	-2	0.5
synthetic how 2	2	2	-1	1	2	1	2	3	1	0	3	0	-1	2	-1	0	3	1	3	2	2.5

	subject 1	subject 2	subject 3	subject 4	subject 5	subject 6	subject 7	subject 8	subject 9	subject 10	subject 11	subject 12	subject 13	subject 14	subject 15	subject 16	subject 17	subject 18	subject 19	subject 20		SUM/20	variance
																						0	
real hello 1	0	1	1	1	2	2	1	2	-1	2	1	1	2	3	1	2	3	0	0	1		1.25	
real hello 1	-1	3	3	1	2	3	-1	2	2	2	3	-1	1	2	-1	1	1	2	1	3		1.4	
real hello 1	-1	1	1	0	2	2	1	2	2	3	3	1	3	3	-1	1	1	2	2	2		1.5	
real hello 1	-2	2	0	1	2	3	1	2	1	3	2	-1	1	2	1	1	1	3	2	2		1.35	
real hello 1	1	2	2	2	2	3	0	2	3	2	2	-1	2	2	2	2	3	3	2	1		1.85	
real hello 1	-2	2	0	-1	2	2	2	3	0	2	3	-1	3	2	2	1	2	3	1	1		1.35	
mean	-0.8333	1.8333	1.1667	0.6667	2.0000	2.5000	0.6667	2.1667	1.1667	2.3333	2.3333	-0.3333	2.0000	2.3333	0.6667	1.3333	1.8333	2.1667	1.3333	1.6667		1.4500	0.8336
variance	1.3667	0.5667	1.3667	1.0667	0.0000	0.3000	1.0667	0.1667	2.1667	0.2667	0.6667	1.0667	0.8000	0.2667	1.8667	0.2667	0.9667	1.3667	0.6667	0.6667		0.8467	
																						0	
real hello 2	1	1	-2	1	1	3	2	-1	-2	-2	-1	-1	0	2	0	1	2	-1	1	-1		0.2	
real hello 2	-2	1	2	0	2	3	1	-1	-3	0	3	-1	0	3	1	-1	-1	1	-1	2		0.45	
real hello 2	0	2	1	1	1	3	-1	-2	-1	2	-1	-2	-1	1	0	2	0	0	0	0		0.25	
real hello 2	1	2	1	-1	1	2	2	-2	-2	2	0	-1	0	0	2	1	1	1	1			0.55	
real hello 2	-1	2	1	-1	1	2	0	-3	0	0	-1	-2	1	2	2	0	2	0	3	0		0.4	
real hello 2	1	1	1	0	1	2	-1	-1	-1	-1	1	-1	1	0	2	1	2	1	1	0		0.5	
mean	0.0000	1.5000	0.6667	0.0000	1.1667	2.5000	0.5000	-1.6667	-1.5000	0.1667	0.1667	-1.3333	0.1667	1.3333	1.1667	0.6667	1.0000	0.3333	0.8333	0.1667		0.3917	1.0417
variance	1.6000	0.3000	1.8667	0.8000	0.1667	0.3000	1.9000	0.6667	1.1000	2.5667	2.5667	0.2667	0.5667	1.4667	0.9667	1.0667	1.6000	0.6667	1.7667	1.2000		1.1700	
																						0	
real smile 1	0	-1	-3	-1	0	-1	1	2	-2	-1	-2	0	2	0	2	1	-1	1	-2	1		-0.2	
real smile 1	-1	-2	-3	1	1	1	-1	-3	0	-2	1	0	-1	-2	-2	1	2	1	1	0		-0.4	
real smile 1	0	-2	-1	0	2	-1	-1	-1	-1	1	-1	-1	-1	-1	-2	1	1	2	1	0		-0.25	
real smile 1	-2	2	0	-1	1	0	0	-2	-2	2	-1	1	0	1	-3	1	-1	2	1	0		-0.05	
real smile 1	0	-1	1	-1	1	-2	-1	1	0	-1	-1	0	-2	2	-3	1	-1	1	0	-1		-0.35	
real smile 1	2	2	3	-1	1	-3	-1	1	1	0	0	1	-2	1	-1	1	-1	1	1	1		0.35	
mean	-0.1667	-0.3333	-0.5000	-0.5000	1.0000	-1.0000	-0.5000	-0.3333	-0.6667	-0.1667	-0.6667	0.1667	-0.6667	0.1667	-1.5000	1.0000	-0.1667	1.3333	0.3333	0.1667		-0.1500	0.4822
variance	1.7667	3.4667	5.5000	0.7000	0.4000	2.0000	0.7000	3.8667	1.4667	2.1667	1.0667	0.5667	2.2667	2.1667	3.5000	0.0000	1.7667	0.2667	1.4667	0.5667		1.7833	
																						0	
real smile 2	1	2	-2	2	1	2	3	2	2	-1	1	0	1	3	0	1	2	1	0	1		1.1	
real smile 2	0	-3	1	1	0	3	3	3	-1	2	2	1	1	3	-1	0	3	2	-1	2		1.05	
real smile 2	0	-2	-2	2	1	2	2	2	2	-1	1	1	0	3	2	1	3	2	0	2		1.05	
real smile 2	1	-3	0	1	2	2	1	1	-1	-2	0	0	-1	3	-3	1	2	2	-1	2		0.35	
real smile 2	0	0	1	1	2	3	2	1	2	-1	1	1	-1	3	-3	2	2	3	1	-3		0.85	
real smile 2	0	0	1	1	1	1	1	3	1	-3	-1	0	-2	1	-1	1	2	1	-2	0		0.25	
mean	0.3333	-1.0000	-0.1667	1.3333	1.1667	2.1667	2.0000	2.0000	0.8333	-1.0000	0.6667	0.5000	-0.3333	2.6667	-1.0000	1.0000	2.3333	1.8333	-0.5000	0.6667		0.7750	1.3721
variance	0.2667	4.0000	2.1667	0.2667	0.5667	0.5667	0.8000	0.8000	2.1667	2.8000	1.0667	0.3000	1.4667	0.6667	3.6000	0.4000	0.2667	0.5667	1.1000	3.8667		1.3850	
																						0	
real hi 1	2	3	3	1	3	3	3	3	3	1	2	1	2	2	3	3	-3	3	2	3		2.15	
real hi 1	0	2	2	1	2	2	1	3	3	2	2	2	2	1	3	3	-3	3	0	2		1.65	
real hi 1	2	2	1	1	2	3	2	2	1	-2	2	1	2	2	3	3	-3	2	3	1		1.5	
real hi 1	-1	2	0	1	2	2	2	3	0	1	3	1	1	1	3	3	-2	3	1	3		1.45	
real hi 1	1	3	2	1	2	1	2	3	2	2	3	1	2	2	3	3	-2	3	2	1		1.85	
real hi 1	2	3	2	2	2	2	3	3	3	2	2	1	1	1	3	2	-2	3	0	3		1.9	
mean	1.0000	2.5000	1.6667	1.1667	2.1667	2.1667	2.1667	2.8333	2.0000	1.0000	2.3333	1.1667	1.6667	1.5000	3.0000	2.8333	-2.5000	2.8333	1.3333	2.1667		1.7500	1.4137
variance	1.6000	0.3000	1.0667	0.1667	0.1667	0.5667	0.5667	0.1667	1.6000	2.4000	0.2667	0.1667	0.2667	0.3000	0.0000	0.1667	0.3000	0.1667	1.4667	0.9667		0.6333	
																						0	
real hi 2	0	2	3	1	1	0	0	3	2	1	2	0	1	0	2	2	-1	2	1	3		1.25	
real hi 2	2	3	2	2	3	0	3	3	3	2	2	2	1	3	3	3	-2	3	2	1		2.05	
real hi 2	2	2	2	1	2	3	3	2	2	-1	3	2	-1	1	3	3	-3	3	2	3		1.7	
real hi 2	1	0	3	1	2	2	1	3	-2	1	2	0	2	0	2	3	-1	3	0	2		1.25	
real hi 2	-1	2	2	1	2	2	1	3	-1	1	1	1	1	1	3	2	-1	3	0	2		1.25	
real hi 2	0	2	2	1	2	3	2	3	3	2	3	1	1	1	3	3	-1	2	-1	2		1.7	
mean	0.6667	1.8333	2.3333	1.1667	2.0000	1.6667	1.6667	2.8333	1.1667	1.0000	2.1667	1.0000	0.8333	1.0000	2.6667	2.6667	-1.5000	2.6667	0.6667	2.1667		1.5333	1.0339
variance	1.4667	0.9667	0.2667	0.1667	0.4000	1.8667	1.4667	0.1667	4.5667	1.2000	0.5667	0.8000	0.9667	1.2000	0.2667	0.2667	0.7000	0.2667	1.4667	0.5667		0.9800	
																						0	
real how 1	-2	0	-2	1	-2	-2	2	2	2	2	0	-1	0	1	3	0	-1	1	3	-3		0.2	
real how 1	0	-1	0	1	-3	-1	2	1	2	2	-2	-1	1	1	2	-1	1	1	1	0		0.3	
real how 1	3	-1	-1	1	-2	-1	0	-1	-1	-1	-1	-1	1	0	-1	0	1	1	0	-2		-0.3	
real how 1	2	1	-1	1	-1	0	1	1	-1	1	-2	-2	2	1	2	0	1	2	2	-3		0.35	
real how 1	-1	2	0	2	-1	-1	1	2	2	0	1	-1	2	2	1	0	1	2	-2	-3		0.45	
real how 1	-1	0	-1	1	-1	2	1	1	1	-1	-1	-2	2	0	2	-1	2	1	2	-2		0.25	
mean	0.1667	0.1667	-0.8333	1.1667	-1.6667	-0.5000	1.1667	1.0000	0.8333	0.5000	-0.8333	-1.3333	1.3333	0.8333	1.5000	-0.3333	0.8333	1.3333	1.0000	-2.1667		0.2083	1.2131
variance	3.7667	1.3667	0.5667	0.1667	0.6667	1.9000	0.5667	1.2000	2.1667	1.9000	1.3667	0.2667	0.6667	0.5667	1.9000	0.2667	0.9667	0.2667	3.2000	1.3667		1.2550	
																						0	
real how 2	0	0	-3	1	-1	1	1	2	1	-1	-1	-2	3	1	-2	-2	0	-1	2	-1		-0.1	
real how 2	0	1	-3	0	-1	2	1	3	0	0	0	-2	2	2	-2	-2	0	-1	1	-3		-0.1	
real how 2	-1	2	-3	2	-1	1	1	2	-1	-1	-2	0	1	1	-1	-2	2	0	2	-3		-0.05	
real how 2	1	2	-3	1	-1	1	-1	3	0	0	-1	-2	1	1	-1	-1	1	-1	1	-2		-0.05	
real how 2	0	2	0	1	-1	1	0	3	0	-1	1	-2	2	1	1	0	2	-1	0	-1		0.4	
real how 2	1	2	-2	0	0	1	2	3	0	-2	-1	-2	2	1	0	-2	2	-1	3	-2		0.25	
mean	0.1667	1.5000	-2.3333	0.8333	-0.8333	1.1667	0.6667	2.6667	0.0000	-0.8333	-0.6667	-1.6667	1.8333	1.1667	-0.8333	-1.5000	1.1667	-0.8333	1.5000	-2.0000		0.0583	1.9920
variance	0.5667	0.7000	1.4667	0.5667	0.1667	0.1667	1.0667	0.2667	0.4000	0.5667	1.0667	0.6667	0.5667	0.1667	1.3667	0.7000	0.9667	0.1667	1.1000	0.8000		0.6750	
Mean of mean	0.1667	1.0000	0.2500	0.7292	0.8750	1.3333	1.0417	1.4375	0.4792	0.3750	0.6875	-0.2292	0.8542	1.3750	0.7083	0.9583	0.3750	1.4583	0.8125	0.3542	0.0000	0.7521	0.2102
mean of variance	1.5500	1.4583	1.7833	0.4875	0.3167	0.9583	1.0167	0.9125	1.9542	1.7333	1.0792	0.5125	0.9458	0.8500	1.6833	0.3917	0.9417	0.4667	1.5292	1.2500	0.0000	1.0910	

	subject 1	subject 2	subject 3	subject 4	subject 5	subject 6	subject 7	subject 8	subject 9	subject 10	subject 11	subject 12	subject 13	subject 14	subject 15	subject 16	subject 17	subject 18	subject 19	subject 20		SUM/20	variance
synthetic smile 1	-1	0	-1	-1	0	1	1	1	-3	0	2	1	1	1	-1	2	2	1	1	2		0.45	
synthetic smile 1	2	-1	3	2	2	3	2	3	0	1	1	1	1	2	-1	1	2	2	0	1		1.35	
synthetic smile 1	0	-1	3	1	2	1	1	2	1	1	2	1	0	2	2	1	1	0	1	0		1.05	
synthetic smile 1	0	1	2	1	1	0	1	-1	-1	-1	1	0	0	2	-3	1	1	1	1	0		0.35	
synthetic smile 1	0	1	-1	1	1	-1	2	1	1	0	1	-1	-2	1	-3	1	1	1	0	1		0.25	
synthetic smile 1	0	-2	2	-1	2	0	2	3	2	1	1	0	-2	2	-3	1	1	2	2	2		0.75	
mean	0.1667	-0.3333	1.3333	0.5000	1.3333	0.6667	1.5000	1.5000	0.0000	0.3333	1.3333	0.3333	-0.3333	1.6667	-1.5000	1.1667	1.3333	1.1667	0.8333	1.0000		0.7000	0.6538
variance	0.9667	1.4667	3.4667	1.5000	0.6667	1.8667	0.3000	2.3000	3.2000	0.6667	0.2667	0.6667	1.8667	0.2667	3.9000	0.1667	0.2667	0.5667	0.5667	0.8000		1.2867	
																						0	
synthetic hello 2	-2	1	0	-1	2	2	0	1	2	-3	2	-1	-1	-2	2	2	1	1	-1	1		0.3	
synthetic hello 2	-3	0	1	-1	1	3	1	-1	1	2	0	-1	-1	-2	1	0	1	0	-3	0		-0.05	
synthetic hello 2	-2	-2	0	-2	1	1	-1	-3	1	2	1	-1	1	-2	0	1	3	0	-1	2		-0.05	
synthetic hello 2	-1	1	1	1	0	2	0	-3	-3	1	-2	-1	2	-2	1	0	2	1	-3	2		-0.05	
synthetic hello 2	-1	2	-1	-1	1	0	0	-1	-2	2	-1	-1	2	-2	-2	0	-2	0	-3	2		-0.4	
synthetic hello 2	-1	0	-1	-1	0	-1	0	1	1	-1	-1	-2	0	-1	-1	1	1	1	-2	1		-0.3	
mean	-1.6667	0.3333	0.0000	-0.8333	0.8333	1.1667	0.0000	-1.0000	0.0000	0.5000	-0.1667	-1.1667	0.5000	-1.8333	0.1667	0.6667	1.0000	0.5000	-2.1667	1.3333		-0.0917	1.0482
variance	0.6667	1.8667	0.8000	0.9667	0.5667	2.1667	0.4000	3.2000	4.0000	4.3000	2.1667	0.1667	1.9000	0.1667	2.1667	0.6667	2.8000	0.3000	0.9667	0.6667		1.5450	
																						0	
synthetic hello 1	2	2	1	0	0	3	0	1	-3	-1	2	-2	0	3	2	0	-2	2	1	-2		0.45	
synthetic hello 1	1	3	1	1	3	2	1	-2	-3	1	2	-2	-1	2	3	2	-3	2	2	2		0.85	
synthetic hello 1	2	2	2	-2	1	3	1	-3	-3	-2	3	-1	-1	1	-1	1	-3	1	2	2		0.25	
synthetic hello 1	2	3	2	-1	2	3	2	-2	-3	-1	3	-1	-1	3	2	1	-3	3	0	1		0.75	
synthetic hello 1	-1	1	-2	-1	2	3	1	-1	-2	-1	3	-1	-1	0	2	1	-3	0	-1	0		-0.05	
synthetic hello 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	
mean	1.0000	1.8333	0.6667	-0.5000	1.3333	2.3333	0.8333	-1.1667	-2.3333	-0.6667	2.1667	-1.1667	-0.6667	1.5000	1.3333	0.8333	-2.3333	1.3333	0.6667	0.5000		0.3750	1.9236
variance	1.6000	1.3667	2.2667	1.1000	1.4667	1.4667	0.5667	2.1667	1.4667	1.0667	1.3667	0.5667	0.2667	1.9000	2.2667	0.5667	1.4667	1.4667	1.4667	2.3000		1.4083	
																						0	
synthetic smile 2	0	0	-2	2	2	2	2	3	2	2	0	0	2	2	1	2	1	2	1	2		1.3	
synthetic smile 2	0	2	0	2	2	-1	0	2	0	3	0	1	-1	-1	2	1	3	2	2	1		1	
synthetic smile 2	0	1	-1	1	0	1	2	3	2	1	0	1	2	1	1	1	3	2	0	1		1.1	
synthetic smile 2	2	2	-1	0	1	0	2	3	2	3	0	0	2	0	-2	1	3	2	1	1		1.1	
synthetic smile 2	2	2	1	1	2	3	3	3	2	-1	0	1	1	1	-1	2	3	2	-1	1		1.35	
synthetic smile 2	1	2	1	2	1	1	2	3	1	1	1	0	0	2	1	1	3	1	1	2		1.35	
mean	0.8333	1.5000	-0.3333	1.3333	1.3333	1.0000	1.8333	2.8333	1.5000	1.5000	0.1667	0.5000	1.0000	0.8333	0.3333	1.3333	2.6667	1.8333	0.6667	1.3333		1.2000	0.5924
variance	0.9667	0.7000	1.4667	0.6667	0.6667	2.0000	0.9667	0.1667	0.7000	2.3000	0.1667	0.3000	1.6000	1.3667	2.2667	0.2667	0.6667	0.1667	1.0667	0.2667		0.9367	
																						0	
synthetic how 1	3	2	2	0	-1	2	-1	3	2	2	0	-1	2	1	3	0	1	0	2	-2		1	
synthetic how 1	-1	1	0	0	-1	3	-2	2	-1	-1	-1	-1	1	0	-1	-1	1	0	0	1		-0.05	
synthetic how 1	-1	1	1	2	-1	-1	-2	1	2	3	-1	-1	1	0	1	0	1	0	0	-3		0.15	
synthetic how 1	-1	1	-2	1	0	3	1	2	2	2	-1	-1	1	1	-1	-1	2	0	0	-1		0.4	
synthetic how 1	1	1	-1	1	0	1	0	1	2	-1	0	-1	2	2	1	0	1	-1	1	-1		0.45	
synthetic how 1	1	2	2	1	0	3	2	3	0	-1	1	-1	1	2	2	1	2	0	1	2		1.2	
mean	0.3333	1.3333	0.3333	0.8333	-0.5000	1.8333	-0.3333	2.0000	1.1667	0.6667	-0.3333	-1.0000	1.3333	1.0000	0.8333	-0.1667	1.3333	-0.1667	0.6667	-0.6667		0.5250	0.7376
variance	2.6667	0.2667	2.6667	0.5667	0.3000	2.5667	2.6667	0.8000	1.7667	3.4667	0.6667	0.0000	0.2667	0.8000	2.5667	0.5667	0.2667	0.1667	0.6667	3.4667		1.3583	
																						0	
synthetic hi 2	-2	1	-1	1	1	1	2	3	-2	2	3	0	2	3	3	1	0	2	2	1		1.15	
synthetic hi 2	-1	2	3	2	2	1	1	2	0	3	3	1	2	2	3	1	0	2	2	3		1.7	
synthetic hi 2	0	1	-2	0	2	1	1	1	-2	0	1	0	1	2	3	3	1	2	2	2		0.95	
synthetic hi 2	0	2	-1	1	2	2	0	-1	-1	1	2	0	2	1	3	2	2	2	0	2		1.05	
synthetic hi 2	-1	2	0	1	2	1	1	2	0	-1	2	-1	1	2	3	1	2	2	2	-2		0.95	
synthetic hi 2	-1	2	1	1	1	3	1	2	0	3	3	-1	2	1	3	1	2	2	1	3		1.5	
mean	-0.8333	1.6667	0.0000	1.0000	1.6667	1.5000	1.0000	1.5000	-0.8333	1.3333	2.3333	-0.1667	1.6667	1.8333	3.0000	1.5000	1.1667	2.0000	1.5000	1.5000		1.2167	0.9652
variance	0.5667	0.2667	3.2000	0.4000	0.2667	0.7000	0.4000	1.9000	0.9667	2.6667	0.6667	0.5667	0.2667	0.5667	0.0000	0.7000	0.9667	0.0000	0.7000	3.5000		0.9633	
																						0	
synthetic hi 1	0	-1	1	0	-1	2	-1	1	1	1	-1	-1	1	2	2	-1	-3	0	1	-2		0.05	
synthetic hi 1	-2	1	-1	1	0	2	2	1	3	3	-1	-1	1	0	1	1	-3	1	1	0		0.5	
synthetic hi 1	2	1	-1	0	-2	0	-1	-2	1	-1	1	-1	-1	1	1	1	0	2	0	1		0.1	
synthetic hi 1	0	2	-2	1	-1	1	2	-3	3	0	-2	-2	-1	2	-1	1	-1	0	2	1		0.1	
synthetic hi 1	3	2	0	-1	-1	3	1	-1	0	-2	-1	-1	1	2	1	1	1	1	1	0		0.5	
synthetic hi 1	0	1	-3	1	-1	2	1	-1	3	1	-3	-1	0	2	1	1	-1	1	2	2		0.4	
mean	0.5000	1.0000	-1.0000	0.3333	-1.0000	1.6667	0.6667	-0.9333	1.8333	0.3333	-1.1667	-1.1667	0.1667	1.5000	0.8333	0.6667	-1.1667	0.8333	1.1667	0.3333		0.2750	0.9950
variance	3.1000	1.2000	2.0000	0.6667	0.4000	1.0667	1.8667	2.5667	1.7667	3.0667	1.7667	0.1667	0.9667	0.7000	0.9667	0.6667	2.5667	0.5667	0.5667	1.8667		1.4250	
																						0	
synthetic how 2	1	2	-3	1	2	2	1	2	0	-1	2	0	1	0	-1	-2	0	1	1	1		0.5	
synthetic how 2	2	2	3	3	2	2	-1	3	2	2	2	2	2	0	3	0	1	1	3	3		1.85	
synthetic how 2	0	1	-1	1	2	0	1	3	2	2	2	1	2	1	2	0	2	2	1	2		1.3	
synthetic how 2	1	2	-2	2	1	2	2	3	3	-1	2	1	1	1	2	1	2	1	1	3		1.4	
synthetic how 2	1	2	-1	1	2	1	1	2	3	-1	2	0	1	1	2	0	2	1	2	1		1.15	
synthetic how 2	2	2	-1	1	2	1	2	3	1	0	3	0	-1	2	-1	0	3	1	3	2		1.25	
mean	1.1667	1.8333	-0.8333	1.5000	1.8333	1.3333	1.0000	2.6667	1.8333	0.1667	2.1667	0.6667	1.0000	0.8333	1.1667	-0.1667	1.6667	1.1667	1.8333	2.0000		1.2417	0.6944
variance	0.5667	0.1667	4.1667	0.7000	0.1667	0.6667	1.2000	0.2667	1.3667	2.1667	0.1667	0.6667	1.2000	0.5667	2.9667	0.9667	1.0667	0.1667	0.9667	0.8000		1.0483	
Mean of mean	0.1875	1.1458	0.0208	0.5208	0.8542	1.4375	0.8125	0.9375	0.3958	0.5208	0.8125	-0.3958	0.5833	0.9167	0.7708	0.7292	0.7083	1.0833	0.6458	0.9167	0.0000	0.6802	
mean of variance	1.3875	0.9125	2.5042	0.8208	0.5625	1.5625	1.0458	1.6708	1.9042	2.4625	0.9042	0.3875	1.0417	0.7917	2.1375	0.5708	1.2583	0.4250	0.8708	1.7083	0.0000	1.2465	

References

[1] J. Ahlberg. Facial feature extraction using eigenspaces and deformable graphs. In *International Workshop on Synthetic/Natural Hybrid Coding and 3-D Imaging*, 1999.

[2] J. Allen, M. S. Hunnicutt, and D. F. Klatt. *From text to speech—the MITalk system.* MIT Press, 1987.

[3] A. M. Baumberg and D. C. Hogg. Learning flexible models from image sequences. In *European Conference on Computer Vision*, pages 299–308. Springer Verlag, 1994.

[4] D. J. Beymer. Face recognition under varying pose. In *AIM*, December 1993.

[5] E. Bienenstock, L. Cooper, and P. Munro. Theory for the development of neuron selectivity; orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2:32–48, 1982.

[6] R. Bowden, P. KaewTraKulPong, and M. Lewin. Jeremiah: The face of computer vision. In *International Symposium on Smart Graphics,*, pages 124–128, 2002.

[7] C. Breaeal. Robot in society: Friend or appliance. In *Agents*, 1999.

[8] N. M. Brooke and S. D. Scott. Computer graphics animations of talking faces based on stochastic models. In *Int Symposium on Speech, Image Processing and Neural Networks*, 1994.

[9] N. M. Brooke, S. D. Scott, and M. J. Tomlinson. Making talking heads and speechreadings with computers. In *Integrated Audio-Visual Processing for Recognition, Synthesis and Communication*, 1996.

[10] R. Brunelli, D. Falavigna, T. Poggio, and L. Stringa. Automatic person recognition by using acoustic and geometric features,. In *Machine Vision and Applications*, volume 8, pages 317–325. M.I.T., 1995.

[11] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Trans on PAMI*, 15:1042–1052, 1993.

[12] J. P. Campbell, T. E. Tremain, and V. C. Welch. The proposed federal standard 1016 4800 bps voice coder: Celp. *Speech Technology Magazine*, pages 58–64, April/May 1990.

[13] J. Cassel, C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, and M. Stone. Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *SIGGRAPH*, 1994.

[14] R. Chellappa, C. L. Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. In *Proceedings of the IEEE*, volume 83, 1995.

[15] H. L. Choong, S. K. Jun, and H. P. Kyu. Automatic human face location in a complex obackground using motion and color information. *Pattern Recognition*, 29:1877–1889, 1996.

[16] R. A. Cole, J. Mariani, H. Uszkoreit, A. Zaenen, and V. Zue. *Survey of the State of the Art in Human Language Technology.* Cambridge University Press, 1996.

[17] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *ECCV*, 1998.

[18] T. F. Cootes and C. J. Taylor. Smart snakes. In *British Machine Vision Conference*, pages 266–275, 1992.

[19] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In *Proc. British Machine Vision Conference*, pages 9–18, 1992.

[20] E. Cosatto and H. P. Graf. Sample-based synthesis of photo-realistic talking heads. In *Computer Animation*, pages 103–110, 1998.

[21] M. Covell. Eigen-points : Control-point location using principal component analyses. In *Faces and Gestures*, pages 122–127, 1996.

[22] J. L. Crowley and F. Berard. Multi-modal tracking of faces for video communications. In *CVPR*, 1997.

[23] J. L. Crowley and J. Martin. Experimental comparison of correlation techniques. In *International Conference on Intelligent Autonomous Systems*, 1995.

[24] J. W. Davis and A. F. Bobick. The representation and recognition of human movement using temporal templates. In *CVPR*, 1996.

[25] D. DeCarlo and D. Metaxas. Optical flow constraints on deformable models with applications to face tracking. In *IJCV*, volume 38, pages 99–127, 2000.

[26] B. DeCarolis, C. Pelachaud, I. Poggi, and F. de Rosis. Behavior planning for a reflexive agent. In *IJCAI*, 2001.

[27] I. Essa and A. Pentland. Coding, analysis, interpretation and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 1995.

[28] K. Etemad and R. Chellappa. Discriminant analysis for recognition of human face images. In *First International Conference of the AVBPA*, pages 127–142, 1997.

[29] T. Ezzat and T. Poggio. Facial analysis and synthesis using image-based models. In *Face and Gesture Recognition*. M.I.T., 1998.

[30] J. Flanagan. Speech coding. *IEEE Transactions on Communications*, 27:710–737, April 1979.

[31] J. L. Flanagan and L. R. Rabiner. *SPEECH SYNTHESIS*. Dowden Hutchinsonand Ross, Inc., 1973.

[32] A. Galata, N. Johnson, and D. C. Hogg. Learning structured behaviour models using variable length markov models. In *IEEE International Workshop on Modelling People*, 1999.

[33] N. J. Gordon, D. J. Salmond, and A. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE-Proceedings-F*, 140(2):107–113, April 1993.

[34] H. P. Graf and E. Cosatto. Face analysis for the synthesis of photo-realistic talking heads. In *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 2000.

[35] R. M. Gray. Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, April 1984.

[36] P. Hancock, V. Bruce, and A. M. Burton. Testing principal component representations for faces. In *4th Neural Computation and Psychology Workshop*, 1997.

[37] O. Hasegawa, C. Lee, W. Wongwarawipat, and M Ishizuka. Realtime synthesis of moving human-like agent in response to user's moving image. In *Pattern recognition*, 1992.

[38] O. Hasegawa, K. Yokosawa, and M. Ishizuka. Real-time parallel and cooperative recognition of facial images for an interactive visual human interface. In *Pattern recognition*, 1994.

[39] A. Hill and C. J. Taylor. Automatic landmark generation for point distribution models. In *Proc. British Machine Vision Conference*, volume 2, pages 429–438, 1994.

[40] E. Hjelmas and J. Wroldsen. Recognizing faces from the eyes only. In *11th Scandinavian Conference on Image Analysis*, 1999.

[41] D. C. Hogg, N. Johnson, R. Morris, D. Bueshing, and A. Galata. Visual models of interaction. In *2nd International Workshop on Cooperative Distributed Vision*, 1998.

[42] G. Holst. Face detection by facets: Combined bottom-up and top-down search using compound templates. In *International Conference on Image Processing*, 2000.

[43] X. W. Hou, S. Z. Li, H. Zhang, and Q. Cheng. Direct appearance models. In *CVPR*, 2001.

[44] X. D. Huang and K. F. Lee. On speaker-independent, speaker-dependent, and speaker-adaptive speech recognition. *IEEE Transactions on Speech and Audio Processing*, 2:150–157, April 1993.

[45] M. Hunt and C. Lefebvre. A comparison of several acoustic representations for speech recognition with degraded and undegraded speech. In *ICASSP*, pages 262–265, 1989.

[46] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *ECCV*, 1996.

[47] T. Jebara and A. Pentland. Maximum conditional likelihood via bound maximization and the cem algorithm. Technical report, M.I.T, University of Massachusetts, 1998.

[48] T. Jebara and A. Pentland. Action reaction learning : Automatic visual analysis and synthesis of interactive behaviour. In *ICVS*, pages 273–292, January 1999.

[49] S. Jen, Y. M. Liao Hong, C. H. Chin, Y. C. Ming, and T. L. Yao. Facial feature detection using geometrical face model: an efficient approach. *Pattern Recognition*, 31:273–282, march 1998.

[50] N. Johnson. *Learning Object Behaviour Models.* PhD thesis, School of Computer Studies, University of Leeds, September 1998.

[51] N. Johnson, A. Galata, and D. C. Hogg. The Acquisition and Use of Interaction Behaviour Models. In *Computer Society Conference on Computer Vision and Pattern Recognition*, June 1998.

[52] N. Johnson and D. C. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14(8):609–615, 1996.

[53] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *First International Conference on Computer Vision*, pages 259–268, 1987.

[54] S. Kawato and J. Ohya. Automatic skin-color distribution extraction for face detection and tracking. In *5th Int. Cong. on Signal Processing*, volume 2, pages 1415–1418, 2000.

[55] S. Kawato and J. Ohya. Automatic skin-color distribution extraction for face detection and tracking. In *4th Int. Cong. on Automoatic Face and Gesture Recognition*, pages 40–45, 2000.

[56] V. Kraft and T. Portele. Quality evaluation of five german speech synthesis systems. In *Acta Acustica*, pages 351–365, 1995.

[57] Z. Liu and Y. Wang. Face detection and tracking in video using dynamic programming. In *International Conference on Image Processing*, 2000.

[58] B. Low and E. Hjelmas. Face detection: A survey. *Computer Vision and Image Understanding*, 83(3):236–274, September 2001.

[59] B. K. Low. *Computer extraction of human faces.* PhD thesis, Department of Electronic and Electrical Engineering, De Montfort University, Leicester, UK, 1998.

[60] J. Makhoul, S. Roucos, and J. Gish. Vector quantization in speech coding. In *Proceeding of the IEEE*, volume 73, pages 1551–1588, 1985.

[61] J. D. Markel and A. H. Gray. *Linear Prediction of Speech.* Springer-Verlag, 1976.

[62] S. McKenna, S. Gong, and H. Liddell. Real-time tracking for an integrated face recognition system. In *2nd Workshop on Parallel Modelling of Neural Operators*, 1995.

[63] S. McKenna, S. Gong, and Y. Raja. Modelling facial colour and identity with gaussian mixtures. *Pattern Recognition*, 12(31):1883–1892, 1998.

[64] B. Menser and M. Brnig. Locating human faces in color images with complex background. In *IEEE Int. Symposium on Intelligent Signal Processing and Communication Systems*, pages 533–536, 1999.

[65] S. Morishima, K. Aizawa, and H. Harashima. An intelligent facial image coding driven by speech and phoneme. In *Acoustic, Speech and Sound processing*, 1989.

[66] R. Morris and D. C. Hogg. Statistical models of object interaction. In *IEEE Workshop on Visual Surveillance*, 1998.

[67] E. Moulines and F. Charpentier. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Communication*, 9:453–468, 1990.

[68] C. Nastar and M. Mitschke. Real-time face recognition using feature combination. In *Faces and Gestures*, 1998.

[69] J. Y. Noh and U. Neumann. A survey of facial modeling and animation techniques. Technical Report 99-705, USC, 1998.

[70] J. Y. Noh and U. Neumann. Talking faces. In *IEEE International Conference on Multimedia and Expo*, 2000.

[71] J. Y. Noh and U. Neumann. Expression cloning. In *SIGGRAPH*, 2001.

[72] A. M. Noll. Cepstrum pitch determination. *Journal of the Acoustical Society of America*, 42:293–309, 1967.

[73] J. O'Connor and J. Seymour. *Introducing neuro-linguistic programming : psychological skills for understanding and influencing people.* Aquarian/Thorsons, 1993.

[74] N. Oliver, A. Pentland, and F. Berard. Lafter: A real-time lips and face tracker with facial expression recognition. In *CVPR*, 1997.

[75] J. Olives, M. Sams, J. Kulju, and O. Seppala. Towards a high quality finnish talking head. In *Multimedia Signal Processing*, 1999.

[76] K. K. Paliwal and B. S. Atal. Efficient vector quantization of lpc parameters at 24 bits/frame. *IEEE Transactions On Speech and Audio Processing*, 1993.

[77] S. Parthasarathy and C. H. Coker. Automatic estimation of articulatory parameters. In *Computer Speech and Language*, volume 6, pages 37–75, 1992.

[78] S. Pasquariello and C. Pelachaud. Greta: A simple facial animation engine. In *6th Online World Conference on Soft Computing in Industrial Applications*, 2001.

[79] C. Pelachaud and I. Poggi. Facial performative in a conversational system. In *Workshop on Embodied Conversational Characters*, 1998.

[80] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing realistic facial expressions from photographs. *Computer Graphics*, 32:75–84, 1998.

[81] F. Plante, G. Meyer, and W. A. Ainsworth. A pitch extraction reference database. In *EUROSPEECH*, pages 837–840, 1995.

[82] I. Poggi and C. Pelachaud. Performative faces. In *Speech Communication*, volume 26, pages 5–21, 1998.

[83] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[84] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg, and C. A. McGonegal. A comparative performance study of several pitch detection algorithms. *IEEE Trans*, 5:399–418, 1976.

[85] M. Reiss and J. G. Taylor. Storing temporal sequences. *Neural Networks*, 4:773–787, 1991.

[86] R. Romano, D. Beymer, and T. Poggio. Face verification for real-time applications. In *Image Understanding Workshop*, pages 747–756. M.I.T., 1996.

[87] B. G. Schunk. Image flow segmentation and estimation by constraint line clustering. *IEEE Transaction PAttern Analysis and MAchine Intelligence*, 11, 1989.

[88] K. Schwerdt. *Compression Video fonde sur l'apparence.* PhD thesis, Laobratoire PRIMA-INRIA de Grenoble, 2001.

[89] K. Schwerdt, J. L. Crowley, and J. B. Durand. Robustification of detection and tracking of faces. In *Joint TMR Workshop on Computer Vision and Mobile Robitics,*, pages 155–161, 1998.

[90] H. Shatkay. The fourier transform - a primer. Technical report, Brown University, Department of Computeur Science, 1995.

[91] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *J Opt Soc Am*, 1987.

[92] A. Smal and P. A. Iyengar. Automatic recognition and analysis fo human faces and facial expressions. *Pattern Recognition*, 25(1):65–77, 1992.

[93] K. Sobottka and I. Pitas. Face localization and facial features extraction based on shape and color information. In *Int. Conf. on Image Porcessing*, September 1996.

[94] M. M. Sondhi. New methods of pitch extraction. *IEEE Transactions on Audio and Electroacoustics*, 16:262–266, 1968.

[95] K. N. Stevens and C. A. Bickley. Constraints among parameters simplify control of klatt formant synthesizer. *Phonetics*, 19:161–174, 1991.

[96] J. Strom, T. Jebara, S. Basu, and A. Pentland. Real time tracking and modeling of faces: An ekf-based analysis by synthesis approach. In *Modelling People Workshop*. M.I.T., 1999.

[97] A. Takeuchi and K. Nagao. Communicative facial displays as a new conversational modality. In *INTERCHI-93*, pages 187–193, 1993.

[98] J. Terrillon, M. N. Shirazi, H. Fukamachi, and S. Akamatsu. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In *Proc of the International Conference on Face and Gesture Recognition*, pages 54–62, 2000.

[99] D. Terzopoulos and R. Szeliski. Tracking with Kalman snakes. *Active Vision*, pages 3–20, 1992.

[100] K. R. Thorisson. Gandalf: An embodied humanoid capable of real-time multimodal dialogue with people. In *First ACM International Conference on Autonomous Agents,*. M.I.T., 1997.

[101] K. R. Thorisson. Real-time decision making in multimodal face-to-face communication. In *ICAA*, 1998.

[102] T. E. Tremain. The government standard linear predictive coding algorithm: Lpc-10. *Speech Technology Magazine*, pages 40–49, April 1982.

[103] J. Triesh. Self-organized integration of adaptive visual cues for face tracking. In *SPIE*, volume 4051, pages 397–406, 2000.

[104] M. Turk and A. Pentland. Eigenfaces for recognition. *J Cognitive Neuroscience*, 3, 1991.

[105] J. Van-Santen, R. Sproat, J. Olive, and J. Hirshberg. *Progress in Speech Synthesis*. Springer-Verlag, 1995.

[106] K. N. Walker, T. F. Cootes, and C. J. Taylor. Automatically building appearance models from image sequences using salient features. In *British Machine Vision Conference*, pages 463–472, 1999.

[107] D. L. Wang and M. A. Arbib. Complex temporal sequence learning based on short-term memory. In *Proceedings of the IEEE*, volume 78, pages 1536–1543, 1990.

[108] M. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images : A survey. In *IEEE Transaction on Pattern Analysis and Machin eIntelligence*, volume 24, pages 34–58, January 2002.

[109] L. Yin and A. Basu. Face model adaptation with active tracking. In *ICIP*, 1999.